



Bridging the Gap: Developing Robust and Ethical AI Systems

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE
DEGREE OF

DOCTOR OF PHILOSOPHY

BY

**SAHEB CHHABRA
(PHD16006)**

Computer Science and Engineering
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

February 2025

Declaration

This is to certify that the thesis titled “Bridging the Gap: Developing Robust and Ethical AI Systems” has been authored by me. It presents the research conducted by me under the supervision of Prof. Mayank Vatsa, Prof. Richa Singh, and Dr. Gaurav Gupta.

To the best of my knowledge, it is an original work, both in terms of research content and narrative, and has not been submitted elsewhere, in part or in full, for a degree. Further, due credit has been attributed to the relevant state-of-the-art with appropriate citations and acknowledgments, in line with established norms and practices.



Saheb Chhabra
Department of Computer Science and Engineering
Indraprastha Institute of Information Technology Delhi
New Delhi, 110020.
Date: 07/26

Declaration Regarding Use of AI in Thesis

I acknowledge that I am fully responsible for the entire content of my thesis, including any sections assisted by online tools, including Artificial Intelligence-based tools. I accept full accountability for any violations of ethical standards in publications arising from the use of such tools.



Signature of PhD Student:

Name of PhD Student: Saheb Chhabra

Roll Number: PhD16006

THESIS CERTIFICATE

This is to certify that the thesis titled **Bridging the Gap: Developing Robust and Ethical AI Systems**, submitted by **Saheb Chhabra**, to the Indian Institute of Technology, Madras, for the award of the degree of **Doctor of Philosophy**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.



Prof. Mayank Vatsa
Thesis Supervisor
Department of CSE
IIIT, New Delhi



Prof. Richa Singh
Thesis Supervisor
Department of CSE
IIIT, New Delhi



Dr. Gaurav Gupta
Thesis Supervisor
MEITY, Govt. of India
New Delhi

Date: February 2025

ACKNOWLEDGEMENTS

The completion of this doctoral thesis represents not just my individual effort, but the collective support, guidance, and encouragement of many remarkable individuals to whom I am deeply indebted.

I wish to express my deepest gratitude to my doctoral advisors, **Prof. Mayank Vatsa**, **Prof. Richa Singh**, and **Dr. Gaurav Gupta**, whose extraordinary guidance and mentorship have been foundational to both this research and my growth as a scholar. Their collective expertise has provided an exceptional framework for my doctoral studies. Prof. Vatsa and Prof. Singh's profound knowledge in Artificial Intelligence and Machine Learning, complemented by Dr. Gupta's pioneering contributions to Digital Forensics, has created a unique interdisciplinary perspective that has fundamentally shaped this research. Their rigorous academic standards, innovative thinking, and dedication to excellence have not only elevated the quality of this work but have also instilled in me the importance of pushing scientific boundaries while maintaining scholarly integrity. Their consistent support, constructive criticism, and intellectual insights have been invaluable in navigating the complex landscape of contemporary technological research.

Special thanks go to my senior colleagues, **Dr. Rohit Keshari**, **Dr. Soumyadeep Ghosh**, **Dr. Aakarsh Malhotra**, **Dr. Akshay Agarwal**, **Dr. Maneet Singh**, and **Dr. Shruti Nagpal**, who created a collaborative and supportive research environment. Their willingness to share their experiences, engage in technical discussions, and provide practical advice has been crucial to overcoming numerous research challenges.

I am particularly grateful to my friends and co-authors, **Dr. Puspita Majumdar** and **Kartik Thakral**, whose collaboration has been instrumental in advancing this research. The countless hours we spent brainstorming ideas and debugging code have not only contributed to this thesis but also led to lasting friendships. Their intellectual contributions, dedication, and ability to provide both technical insights and moral support have enriched this work immeasurably. Our academic partnership has evolved into a

meaningful friendship that I deeply value.

I would also like to express my sincere appreciation to all my labmates for creating an intellectually stimulating and supportive environment. The daily discussions over tea, technical brainstorming sessions, and mutual encouragement have made this journey both enlightening and enjoyable. Their collaborative spirit have made our laboratory a second home, fostering both academic growth and lasting friendships.

I would like to extend my heartfelt gratitude to **IIT-Delhi** for providing exceptional infrastructure, which stands as the finest I have encountered in any academic institution. I am deeply thankful to all the faculty members and staff for their unwavering support throughout my Ph.D. journey. I would like to acknowledge **Priti Ma'am** for her assistance with administrative matters and scholarship-related queries and requests. I am also profoundly grateful to my Ph.D. monitoring committee members, **Dr. Saket Anand** and **Dr. Pushpendra Singh**, for their invaluable critical comments and constructive feedback. Additionally, I express my sincere thanks to **Dr. Puneet Goyal** for his valuable insights during my Ph.D. comprehensive examination. Furthermore, I extend my appreciation to **IIT Jodhpur** for providing excellent facilities and support during my tenure as a visiting research scholar.

Finally, I would like to express my heartfelt gratitude to my family, whose unconditional love and support have sustained me throughout this academic pursuit. Their patience, understanding, and encouragement during challenging times have been my foundation.



Saheb Chhabra

ABSTRACT

Machine learning and deep learning algorithms have demonstrated exceptional performance across various computer vision tasks, including face detection, attribute prediction, object detection, image segmentation, and classification. The capabilities of deep neural networks (DNNs) have driven the development of numerous computer vision-based artificial intelligence (AI) systems, employed in diverse applications such as autonomous vehicles, COVID-19 detection in X-ray images, facial recognition for security and surveillance systems. Despite these advancements, the real-world applicability of such systems remains a critical challenge due to issues such as the unethical exploitation of AI technologies and failures of deep models on real-world datasets.

One significant concern is the unethical exploitation of AI systems, particularly in contexts involving privacy violations and the generation of fake media. Instances such as unauthorized profiling for targeted advertising and the creation and dissemination of Deepfake content pose serious ethical and social risks. These activities not only infringe on individual privacy but also create widespread societal distress, potentially leading to psychological harm. Another challenge lies in the failure of deep models to generalize effectively to unseen real-world datasets. This is largely due to a lack of robustness and generalizability in model training, as well as data distribution shifts during testing. For example, a model trained on the Labeled Faces in the Wild (LFW) dataset containing over 13,000 unconstrained facial images collected from the web, may perform poorly when evaluated on the CelebA dataset, which consists of more than 200,000 celebrity face images annotated with 40 facial attributes. This performance drop occurs because of discrepancies in data characteristics.

To address these challenges, this dissertation investigates the problems in the input space, the model space, and the feature space, identifying the input space as the root cause. The lack of a control mechanism to guide the attention of deep models toward meaningful image features underpins these issues. To resolve this, we propose novel solutions leveraging the dynamics of attention mechanisms in deep models.

Two solutions are introduced to mitigate the unethical exploitation of AI systems:

1) Privacy Preservation: A framework that enables users to selectively anonymize facial identity and attribute information in images. This allows users to control the attention of the model, blocking it from processing specific attributes such as gender or age.

2) Deepfake Detection: An approach tailored to detect fake media across varying compression levels and algorithms used on social media platforms. This solution improves detection accuracy by training models to identify unseen artifacts in low-quality compressed media.

The latter part of this dissertation addresses deep model failures on real-world datasets with two innovative solutions:

1) Data Shift Mitigation: A method to counteract data distribution shifts by learning uniform perturbations in the dataset. These perturbations improve model attention toward meaningful features, ensuring better performance in real-world scenarios.

2) Training Robustness: A strategy to diversify attention of the model during training, compelling it to learn overlooked features within the dataset. This approach enhances the robustness and generalizability of the model by expanding the range of features it considers.

Overall, this dissertation presents novel mechanisms to address critical challenges in AI system reliability and ethical deployment by focusing on privacy preservation, Deepfake detection, and generalizable learning. These solutions contribute significantly to the development of robust, generalizable, and ethical AI systems for real-world applications.

Contents

ACKNOWLEDGEMENTS	i
ABSTRACT	iii
LIST OF TABLES	xii
LIST OF FIGURES	xviii
ABBREVIATIONS	xix
1 INTRODUCTION	2
1.1 Unethical Exploitation of AI Systems	2
1.2 Failure of AI Systems	4
1.3 Focus of the Dissertation	6
1.4 Research Contribution	10
1.5 Organization of the Thesis	12
2 Anonymizing k-Facial Attributes via Adversarial Perturbations	13
2.1 Introduction	13
2.1.1 Literature Review	14
2.1.2 Research Contributions	17
2.2 Proposed Approach	18
2.3 Datasets and Experimental Details	21
2.4 Performance Evaluation	23
2.4.1 Multiple Attribute Anonymization	23
2.4.2 Attribute Suppression with Identity Preservation	24
2.5 Robustness Against Adversarial Defenses	27
2.6 Summary	27
3 PrIdentity: Generalizable Privacy Preserving Adversarial Perturbations for Anonymizing Facial Identity	28

3.1	Introduction	28
3.2	Related Work	30
3.3	Privacy Preserving Identity Anonymization	36
3.3.1	Data Utility vs Privacy	38
3.3.2	Generalizability	39
3.4	Experimental Setup	40
3.4.1	Implementation Details	44
3.4.2	Fair Comparison with Existing Methods	44
3.5	Performance Evaluation	45
3.5.1	Single Image Anonymization	46
3.5.2	Multiple Image Anonymization	50
3.5.3	Bounding Box Prediction	51
3.5.4	User Study	52
3.6	Identity Drift in Feature Space	53
3.7	Summary	53
4	Low Quality Deepfake Detection via Unseen Artifacts	54
4.1	Introduction	54
4.2	Related Work	56
4.3	Proposed Algorithm	60
4.3.1	Learning Visibility Matrix for Unseen Imperceptible Artifacts	61
4.3.2	Training with Unseen Imperceptible artifacts	63
4.4	Datasets and Experimental Details	64
4.4.1	Implementation Details	66
4.5	Experimental Results and Analysis	68
4.5.1	Intra-Compression	68
4.5.2	Inter Compression	71
4.6	Model Robustness to Image Degradations	79
4.7	Summary	79
5	Data Fine-tuning	80
5.1	Introduction	80
5.1.1	Related Work	82

5.2	Proposed Approach: Data Fine-tuning	83
5.3	Datasets Protocol and Experimental Details	88
5.3.1	Implementation Details	89
5.4	Performance Evaluation	91
5.4.1	Black Box Data Fine-tuning: Intra Dataset Experiment . . .	91
5.4.2	Black box Data Fine-tuning: Inter Dataset Experiment . . .	92
5.4.3	Model Fine-tuning versus Data Fine-tuning	94
5.4.4	Robustness against New Biases and Distortions	95
5.5	Model Transferability of Learned Perturbation	95
5.6	Summary	96
6	Feature Guided Perturbation for Facial Attribute Classification	97
6.1	Introduction	97
6.2	Literature Review	99
6.2.1	Feature Space Optimization	99
6.2.2	Perturbation Learning	100
6.2.3	Facial Attribute Classification	102
6.3	Proposed Approach	103
6.4	Experiments and Results	108
6.4.1	Datasets & Protocol	109
6.4.2	Implementation Details	110
6.4.3	t-SNE Visualization	111
6.4.4	Single Attribute Classification	112
6.4.5	Multiple Attributes Classification	117
6.4.6	Comparison with Existing Algorithms	122
6.4.7	Choice of Hyperparameters	122
6.4.8	Effectiveness of Feature Drop	123
6.4.9	Robustness against New Biases and Distortions	123
6.5	Summary	124
7	DIVINE: Diverse-Inconspicuous Feature Learning to Mitigate Abridge Learning	125
7.1	Introduction	125

7.2	Related Work	127
7.3	Diverse-Inconspicuous Feature Learning	129
7.3.1	Identification of Inconspicuous Input Features	131
7.3.2	Diverse Unified Model Learning	133
7.4	Experimental Setup	134
7.4.1	Comparison with Existing Approaches	135
7.4.2	Evaluation Metrics	136
7.4.3	Implementation Details	137
7.4.4	Selection Criteria for Number of Feature Maps	139
7.4.5	Implementation Details	140
7.5	Results and Analysis	140
7.5.1	Evaluation on Feature-suppressed Datasets	140
7.5.2	Evaluation on Corruptions and Perturbations	144
7.5.3	Ablation Experiment to Visualize the Trend of Parameter p	150
7.5.4	Computational Runtime	150
7.6	Limitations	151
7.7	Summary	151
8	Conclusion	153
8.1	Future Research Directions	155

List of Tables

2.1	Summarizing the attribute suppression/anonymization algorithms in the literature.	14
2.2	The experiments are performed pertaining to three cases to showcase the effectiveness of the proposed algorithm.	22
2.3	Confusion matrix displaying classification accuracies (%) of before and after suppression of three attributes together on the CelebA dataset .	22
2.4	Comparison of confusion matrix of the proposed algorithm with Mirjalili and Ross (2017) on the MUCT dataset.	25
3.1	Comparison of existing algorithms for identity anonymization. . . .	32
3.2	Details of the experiments performed to evaluate the performance of the proposed algorithm.	38
3.3	Identification accuracy (%) for single image anonymization experiment on the original and identity anonymized images obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm on the CelebA dataset. To showcase the genrealizability of the proposed algorithm, the results are reported on VGGFace, ArcFace, and LightCNN-29 models respectively. A low accuracy indicates a high performance (better privacy).	41
3.4	Data utility (SSIM) scores for single image anonymization experiment on the original and corresponding identity anonymized images obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm on the CelebA dataset. A high score indicates high performance.	42
3.5	Identification accuracy (%) for multiple image anonymization experiment on the original and identity anonymized images obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm on the CelebA dataset. To showcase the generalizability of the proposed algorithm, the results are reported on VGGFace, ArcFace, and LightCNN-29 models respectively. A low accuracy indicates a high performance (better privacy).	44
3.6	Verification performance of the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm at FPR 0.001 on the LFW dataset.	46
3.7	Comparing the True Positive Rate of the proposed algorithm with state-of-the-art identity anonymization algorithms on the LFW dataset at 0.001 False Positive Rate.	48
3.8	Comparing the proposed algorithm with state-of-the-art identity anonymization algorithms on the CelebA-HQ dataset for bounding box distance prediction using MtCNN.	52

4.1	Details of the experiments to showcase the effectiveness of the proposed algorithm.	65
4.2	Classification accuracy (%) on FaceForensics and FaceForensics++ dataset for intra compression experiment at c-40 (hard compression).	66
4.3	Classification accuracy (%) on c-0, c-23, and c-40 videos for inter compression experiment on the FaceForensics dataset.	68
4.4	Classification accuracy (%) on c-23, and c-40 videos when the model is trained on c-0 videos for inter-compression experiment on the FaceForensics++ dataset. The classification accuracy corresponds to Deepfake (DF), Face2Face (F2F), and FaceSwap (FS) alterations with XceptionNet and ResNet50 as a base model.	69
4.5	Classification accuracy (%) on c-0, and c-40 videos when the model is trained on c-23 videos for inter-compression experiment on the FaceForensics++ dataset. The classification accuracy corresponds to Deepfake (DF), Face2Face (F2F), and FaceSwap (FS) alterations with XceptionNet and ResNet50 as a base model.	69
4.6	Classification accuracy (%) on c-0, and c-23 videos when the model is trained on c-40 videos for inter-compression experiment on the FaceForensics++ dataset. The classification accuracy corresponds to Deepfake (DF), Face2Face (F2F), and FaceSwap (FS) alterations with XceptionNet and ResNet50 as a base model.	70
4.7	Mean NIQE quality scores on the FaceForensics++ dataset corresponding to c-0, c-23, and c-40 compressions, respectively. Lower scores represent better quality.	71
4.8	Classification accuracy (%) on c-0, c-23, and c-40 videos corresponding to the proposed algorithm for the inter-compression experiment on the Celeb-DF dataset.	74
4.9	AUC scores (%) on different compression factors on the Celeb-DF dataset. The scores are reported for cross dataset+inter-compression experiment. The model is trained on the FaceForensics++ dataset and evaluated on the Celeb-DF dataset.	77
5.1	Details of the experiments to show the efficacy of the proposed data fine-tuning for facial attribute classification.	88
5.2	Classification accuracy (in %) of before and after Data Fine-tuning(DFT) for ‘Gender’ attribute on LFW, CelebA, and MUCT datasets.	89
5.3	Classification accuracy (in %) before and after performing data fine-tuning for three attributes on the LFW and CelebA datasets.	89
5.4	Confusion matrix of the LFW dataset for three attributes: ‘Smiling’, ‘Bushy Eyebrows’, ‘Pale Skin’.	90
5.5	Classification accuracy(%) of Black box Data Fine-tuning: Inter Dataset experiment for ‘Gender’ attribute on the MUCT, LFW, and CelebA datasets.	92

5.6	Classification accuracy(%) of Black box Data Fine-tuning: Inter Dataset experiment.	92
6.1	Summarizing the experiments performed to evaluate the proposed algorithm for facial attribute classification.	109
6.2	Classification accuracy (%) on five attributes obtained using original and FGP data corresponding to Without MFT.	109
6.3	Classification accuracy (%) on five attributes corresponding to MFT Type I and comparison with the results of DFT Chhabra <i>et al.</i> (2019).	110
6.4	Classification accuracy (%) using original and FGP data on five attributes corresponding to MFT Type II.	114
6.5	Comparing the number of parameters and computational time with mean classification accuracy (%) corresponding to the five attributes of the CelebA and LFW datasets using the original and FGP data.	115
6.6	Comparing the performance with pre-trained VGGFace model on multiple attributes (combination of two and three attributes).	115
6.7	Classification accuracy (%) on multiple attributes (combination of two and three attributes) corresponding to MFT Type I.	116
6.8	Classification accuracy (%) on multiple attributes (combination of two three attributes) corresponding to MFT Type II.	117
6.9	Comparison of the classification accuracy (%) of the proposed algorithm with FLDA on the LFW and CelebA datasets.	118
6.10	Analyzing the effect of varying batch size at learning rate = 0.01 and α = 2.0 on the LFW dataset.	118
6.11	Analyzing the effect of varying learning rate at batch size = 200 and α = 2.0 on the LFW dataset.	119
6.12	Analyzing the effect of varying α at batch size = 200 and learning rate = 2.0 on the LFW dataset.	120
6.13	Classification accuracy (%) of the proposed algorithm with feature drop (best performance), median, and interquartile range corresponding to MFT: Type I and MFT: Type II on the LFW dataset and CelebA dataset.	121
7.1	Classification accuracy (%) of existing and the DIVINE algorithm on the original and feature-suppressed datasets.	138
7.2	Comparison of relative corruption error obtained using existing and proposed methods for each corruption type on the CIFAR10-C and CIFAR100-C dataset.	142
7.3	Classification accuracy (%) obtained using Jacobian Regularization (JR), Random Suppression (RS), and DIVINE algorithm on the TinyImageNet-C dataset for different corruptions.	144

7.4	Classification accuracy (%) obtained using Abridged Learning and DIVINE algorithm on the CIFAR10-P and CIFAR100-P datasets for different perturbations.	145
7.5	Classification accuracy (%) obtained using Abridged Learning and DIVINE algorithm on the TinyImageNet-P dataset for different perturbations.	148
7.6	Comparison of Mean Flip Rate (mFR) of DIVINE with existing algorithms on the CIFAR100-P dataset.	149
7.7	Training time of Abridged Learning and proposed DIVINE method on CIFAR10 dataset.	150

List of Figures

1.1	Illustrating the process of attribute prediction without and with the control mechanism. (a) shows the lack of control mechanism for attribute anonymization, and (b) shows the system with a control mechanism for attribute anonymization where the user can selectively suppress and preserve the attribute.	5
1.2	Illustrating the process of learning of the model. (a) shows the model ignored the relevant features and learned only dominant features (b) shows the model learned both dominant and inconspicuous features.	7
1.3	Illustrating the performance of the model due to data shift. (a) shows the poor performance of the model due to data shift. (b) illustrates the algorithm to shift the data without updating the model.	8
2.1	Profiling of a person using his face image in ID card for malicious purpose.	14
2.2	Demonstrating the attribute anonymized images generated by existing algorithms. First and third column images are original images while second and fourth column images are gender anonymized images (Sim and Zhang (2015), Suo <i>et al.</i> (2011), Othman and Ross (2014), Mirjalili and Ross (2017)).	15
2.3	Illustrating the steps involved in the proposed algorithm.	17
2.4	This figure shows the images before and after anonymizing attributes of Celeb-A dataset. First column represents the original images while second, third, and fourth columns show one, three, and five attributes anonymized images.	22
2.5	Comparing the attribute class score distributions of images before and after anonymization of 5 attributes together on the CelebA dataset. The first row distributions pertain to the original images whereas, second row distributions corresponds to the anonymized images.	24
2.6	Comparing score distributions before and after anonymizing 'Gender' attribute on the CelebA dataset.	24
2.7	Examples of before and after anonymization of 'Gender' attribute on MUCT and LFWcrop datasets. First row images are original images, second row images are 'gender' anonymized images.	25
2.8	CMC curve showing original to original face matching vs original to anonymized face matching on the MUCT dataset.	26
2.9	ROC curve showing face recognition performance on the LFWcrop Dataset	26

3.1	Illustration of information profiling of unknown identity from social media platforms.	29
3.2	Sample identity-anonymized images are taken from the existing literature. In each pair, the image on the left represents the original, while the one on the right represents the anonymized version. “Images in (a) are from Gafni <i>et al.</i> (2019), images in (b) are generated through CIAGAN (Maximov <i>et al.</i> , 2020), images in (c) are generated through RiDDLE (Li <i>et al.</i> , 2023a), images in (d) are generated through FALCO (Barattin <i>et al.</i> , 2023), images in (e) are generated through FIT (Gu <i>et al.</i> , 2020), and images in (f) are generated through G ² Face (Yang <i>et al.</i> , 2024a).	31
3.3	Illustrating the pipeline of the proposed algorithm. (a) shows the training process for learning a perturbation to generate an identity anonymized image. (b) shows that the pre-trained model is unable to extract the identity information from the anonymized image. (c) shows the addition of learned perturbation to obtain identity anonymized image.	33
3.4	Comparison of trade-off score (T) for single image anonymization experiment corresponding to gallery size 3 on the CelebA dataset.	43
3.5	Sample images before and after anonymization obtained using the proposed algorithm $\mathcal{L}_p$ norm.	46
3.6	Comparison of the anonymized images generated using existing algorithms Gafni <i>et al.</i> (2019), Maximov <i>et al.</i> (2020) and the proposed algorithm. The images generated using existing algorithms are heavily distorted while the visual appearance of the images generated using the proposed algorithm is preserved.	48
3.7	Sample images before and after anonymization obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm on the CelebA dataset.	49
4.1	The Class Activation Maps (CAMs) shown in (a) depict that the conventional algorithms fail to focus properly after compression while the focus of the proposed algorithm stays consistent before and after compression. (b) shows that the proposed algorithm performs significantly better when trained on raw videos and tested on raw and compressed videos.	55
4.2	Samples from FaceForensics++ dataset manipulated using different techniques at easy, hard and no compression.	57
4.3	Illustrating the steps involved in the proposed algorithm. (a) The visibility matrix v is learned over a randomly sampled batch of images to enhance the visibility of unseen artifacts. (b) A new dataset with enhanced visibility of artifacts is created and augmented with the original dataset. The model is re-trained over the new dataset. Steps (a) and (b) are performed iteratively.	60

4.4	Samples with highlighted artifacts corresponding to Face2Face alteration on the FaceForensics++ dataset. (a) shows the samples with visually imperceptible artifacts in the bounding box. (b) shows the visualization of the visibility matrices in the jet colormap at step 1 and step 2.	72
4.5	UMAPs depicting the separation between real and fake images in c-40 videos. The fine-tuned and proposed models are trained on c-0 and c-23 individually.	73
4.6	ROC curves comparing the performance of the fine-tuned model and the proposed algorithm on c-40 videos. The models are trained using c-0 videos with XceptionNet as the base model.	73
4.7	Class Activation Maps for some samples that are misclassified as real by the fine-tuned model but correctly classified by the proposed model. (a) is original image classified as Deepfake while (b) and © are Deepfake images classified as original.	75
4.8	Samples misclassified by the proposed algorithm on the FaceForensics++ dataset.	75
4.9	Illustration of Deepfakes publicly circulated on different social media platforms. Typically, on social media platforms, videos undergo different levels of compression. The proposed algorithm successfully identifies the above in-the-wild samples to be fake.	76
5.1	Illustration of model fine-tuning and data fine-tuning: (a) represents the data distribution with two classes. (b) represents Model Fine-tuning where the model’s decision boundary shifts corresponding to the input data, and (c) represents Data Fine-tuning where the input data shifts corresponding to model’s decision boundary.	81
5.2	Comparing the concept of adversarial perturbation with data fine-tuning. (a) Adversarial perturbation: shows the application of perturbation in attacking deep learning models (Xie <i>et al.</i> , 2017). (b) Privacy preservation: perturbation can be used to anonymize the attributes by preserving the identity of the input image (Chhabra <i>et al.</i> , 2018). (c) Data Fine-tuning: illustrates the proposed application of perturbation in enhancing the performance of a model.	82
5.3	(a) Block diagram illustrating the steps of the proposed algorithm. In the first step, perturbation is initialized with zero image and added to the original training data. In the next step, perturbed training data is given as input to the (attribute prediction) model followed by the computation of loss. After that, optimization is performed over perturbation and added to the training data. (b) Some samples of the learned perturbation using the proposed algorithm. The first two visualizations correspond to the perturbation learned for ‘Smiling’ attribute of LFW and CelebA datasets, respectively. The third visualization corresponds to the ‘Gender’ attribute of the MUCT dataset.	84

5.4	Illustration of the proposed DFT algorithm. Figure (a)-(b) represents the training of attribute prediction model using dataset D_1 . (c)-(d) shows the performance of the trained attribute prediction model on dataset X . (e)-(f) shows the performance of the fine-tuned dataset Z by adding perturbation on trained attribute prediction model.	87
5.5	Misclassified samples that are correctly classified after data fine-tuning. First row shows the images misclassified before data fine-tuning while the second row represents their correct classification after data fine-tuning. The first block of images correspond to the ‘Smiling’ attribute, second block corresponds to ‘Bushy Eyebrows’, while the third block corresponds to ‘Pale Skin’ of the LFW dataset.	87
5.6	Smiling attribute score distribution pertaining to before and after performing data fine-tuning on the LFW dataset. The left graph represents the score distribution before data fine-tuning and right graph represents the score distribution after data fine-tuning.	90
5.7	ROC plots showing before and after data fine-tuning results of Black box Data Fine-tuning: Inter Dataset Experiment. First three ROC curves shows the result on the LFW dataset using a model trained on the CelebA dataset. Last three ROC curves shows the result on the CelebA dataset using a model trained on the LFW dataset.	91
5.8	Score distributions pertaining to before and after performing data fine-tuning. Top three graphs from the left represent the distribution of the LFW dataset predicted using a model trained on the CelebA dataset. Bottom three graphs from the left represent its corresponding distribution after data fine-tuning. Similarly top three graphs from the right represent the score distribution on the CelebA dataset predicted using a model trained on the LFW dataset. Bottom three graphs from the right represent its corresponding distribution after data fine-tuning.	93
5.9	Comparing Data Fine-tuning versus Model Fine-tuning for ‘Gender’ attribute on the MUCT and LFW datasets using a model trained on the CelebA dataset.	94
5.10	Comparing the results of Data Fine-tuning versus Model Fine-tuning on the LFW dataset using a model trained on the CelebA dataset. . .	94
6.1	Illustrating the concept of data shift in the input space on a synthetically generated 2D dataset. (a) Demonstrates the model training on <i>Dataset A</i> with three classes. (b) Shows <i>Dataset A</i> with decision boundaries in the input space corresponding to the pre-trained model and t-SNE representation of the features obtained using the pre-trained model. (c) A new two-class <i>Dataset B</i> with unseen decision boundaries in the input space along with t-SNE representation corresponding to the pre-trained model (on <i>Dataset A</i>). (d) Shows the Feature Guided Perturbed (FGP) dataset obtained after adding the learned perturbation to the <i>Dataset B</i> mentioned in (c) along with its t-SNE representation.	98

6.2	Illustration of the proposed algorithm. (a) Represents the steps involved in learning the perturbation to be added to the input data for obtaining FGP data. (b) Represents the feature space obtained using original and FGP data.	101
6.3	t-SNE visualization corresponding to the ‘Smiling’ attribute of the LFW dataset along with the centers computed using conventional mean and feature drop with 0.1 drop rate.	106
6.4	t-SNE visualization obtained using pre-trained VGGFace model pertaining to the original and FGP data corresponding to the ‘Smiling’ and ‘Bushy Eyebrows’ attributes of the LFW dataset.	110
6.5	Standard deviation of five fold experiments corresponding to MFT Type I on the (a) LFW and (b) CelebA datasets.	111
6.6	Score distribution of ‘Gender’ attribute pertaining to the original and FGP data of the MUCT dataset.	112
6.7	Perturbation visualization corresponding to the ‘Bags Under Eyes’, ‘Pale Skin’, and ‘Oval Face’ attributes of the LFW dataset at different iterations during training.	113
6.8	Standard deviation of five fold experiments corresponding to MFT Type II on the (a) LFW and (b) CelebA datasets.	114
6.9	Summarizing the results of the proposed algorithm and state-of-the-art algorithm on the LFW dataset with 40 attributes.	122
7.1	Illustration of the <i>Abridge Learning</i> (AL) and the proposed method. Model obtained with AL learns only ‘paw’ input feature and ignore other features. This results in failure of the model on the image with missing ‘paw’ feature. The Proposed DIVINE learns ‘paw’ along with feature ‘ear’ as well. This results in successful classification of the image where ‘paw’ feature is missing and the ‘ear’ feature is present.	126
7.2	Pipeline of the proposed diverse and inconspicuous feature learning method. (a) Shows the steps to identify the inconspicuous input features. In steps 1 and 2, the model is trained with original images by minimizing cross-entropy loss function. Steps 3 and 4 shows the computation of the dominance matrix corresponding to each image. In step 5, feature-suppressed dataset is obtained via the dominance matrix and dominance feature maps. In the final step, the feature-suppressed dataset is used for training the model to identify the inconspicuous input features. (b) Shows the unified diverse model training with original images.	130
7.3	Sample original images and the corresponding feature-suppressed images on the MNIST dataset obtained using the proposed method.	137
7.4	Sample original images and the corresponding feature-suppressed images on the CIFAR10 dataset obtained using the proposed method.	139

7.5	Visualizations of the semantically relevant features learned by the model. (a) shows the strokes learned by the model, which distinguishes digit 8 from 2 and (b) shows the feature-suppressed image. Similarly (c) and (d) show the distinguishing strokes learned by the model and the feature-suppressed image.	141
7.6	Illustration of dominant features and inconspicuous features obtained in the MNIST dataset. Sample original images x and the corresponding dominance matrices $D_1(x)$, $D_2(x)$, $D_3(x)$, and $D_u(x)$. $D_1(x)$ and $D_u(x)$ are obtained corresponding to the AL and unified model trained on the original dataset. In contrast, $D_2(x)$ and $D_3(x)$ are obtained corresponding to the model trained on feature-suppressed datasets. . . .	146
7.7	Sample images corrupted with impulse noise and the corresponding dominance matrices obtained using Jacobian regularization, random suppression, and the proposed unified model.	147
7.8	Plot representing the trend line of the average classification accuracy corresponding to baseline and proposed algorithm on different values of p on the MNIST dataset.	147

ABBREVIATIONS

AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
CV	Computer Vision
CNN	Convolutional Neural Network
DNN	Deep Neural Network
COTS	Commercial-of-the-Shelf
FAR	False Accept Rate
ROC	Receiver Operating Characteristic curve
SSIM	Structural Similarity Index
TAR	True Accept Rate
DFT	Data Fine-tuning
FGP	Feature Guided Perturbation
VGGFace	Visual Geometry Group Face
CelebA	Celebrities Attributes Dataset
LFW	Labeled Faces in the Wild
MFT	Model Fine-tuning
DIVINE	Diverse and Inconspicuous Learning
mFR	Mean Flip Rate
CIFAR	Canadian Institute for Advanced Research (datasets)
AL	Abridged Learning
CAM	Class Activation Map
OOD	Out-of-Distribution
GAN	Generative Adversarial Network

PUBLICATIONS

Publications Related to the Dissertation

1. **S. Chhabra**, R. Singh, and M. Vatsa, “**Anonymizing k -Facial Attributes via Adversarial Perturbations**”, In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.
2. **S. Chhabra**, P. Majumdar, R. Singh, and M. Vatsa, “**Data Fine-tuning**”, In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 33, no. 01, pp. 8223-8230. 2019.
3. A. Malhotra, **S. Chhabra**, R. Singh, M. Vatsa “**On Privacy Preserving Anonymization of Finger-selfies**”, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 26-27, 2020.
4. **S. Chhabra**, P. Majumdar, R. Singh, M. Vatsa, “**Feature Guided Perturbation for Facial Attribute Classification**”, *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 6, pp. 1739-1751, 2022.
5. **S. Chhabra**, K. Thakral, S. Mittal, R. Singh, and M. Vatsa, “**Low Quality Deep-Fake Detection via Unseen Artifacts**”, *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 4, pp. 1573-1585, 2023.
6. R. Singh, M. Vatsa, K. Thakral, S. Mittal, **S. Chhabra**. “**System, Apparatus, and Method to Detect Deepfakes**”, *Indian Patent Granted, Patent No: 544678*, 2024. [Patent]
7. **S. Chhabra**, K. Thakral, R. Singh, and M. Vatsa, “**PrIdentity: Generalizable Privacy Preserving Adversarial Perturbations for Anonymizing Facial Identity**”, *IEEE Transactions on Biometrics, Behavior, and Identity Science (TBIOM)*, 2025 (Minor Revision).
8. **S. Chhabra**, K. Thakral, M. Vatsa, and R. Singh, “**DIVINE: Diverse-Inconspicuous Feature Learning to Mitigate Abridge Learning**”, *IEEE Transactions on Machine Learning Research*, 2025 (Accepted).

Other Publications

1. **S. Chhabra**, G. Gupta, M. Gupta, G. Gupta, “**Detecting Fraudulent Bank Checks**”, In *Advances in Digital Forensics XIII: 13th IFIP WG 11.9 International Conference*, pp. 245-266, 2017.

2. P. Majumdar, **S. Chhabra**, R. Singh, and M. Vatsa, “**On Detecting Domestic Abuse via Faces**”, In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 2173-2179, 2018.
3. P. Majumdar, **S. Chhabra**, R. Singh, and M. Vatsa, “**Subclass Contrastive Loss for Injured Face Recognition**”, *IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1-7, 2019.
4. R. Verma, J. Govindaraj, **S. Chhabra**, G. Gupta, “**DF 2.0: An Automated, Privacy Preserving, and Efficient Digital Forensic Framework that leverages Machine Learning for Evidence Prediction and Privacy Evaluation**”, *Journal of Digital Forensics, Security and Law* 14, no. 2, p.3, 2019.
5. **S. Chhabra**, A. Agarwal, R. Singh, M. Vatsa, “**Attack Agnostic Adversarial Defense via Visual Imperceptible Bound**”, In *25th International Conference on Pattern Recognition (ICPR)*, pp. 5302-5309, 2021.
6. P. Majumdar, **S. Chhabra**, R. Singh, M. Vatsa, “**Injured Face Recognition via SCIFI Loss**”, *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 3, no. 1, pp. 112-123, 2020.
7. P. Majumdar, **S. Chhabra**, R. Singh, M. Vatsa, “**Subgroup Invariant Perturbation for Unbiased Pre-Trained Model Prediction**”, *Frontiers in Big Data*, 3, p. 590296, 2021.
8. R. Keshari, S. Ghosh, **S. Chhabra**, M. Vatsa, R. Singh “**Unravelling Small Sample Size Problems in the Deep Learning World**”, *IEEE Sixth International Conference on Multimedia Big Data (BigMM)*, pp. 134-143, 2020.
9. **S. Chhabra**, P. Majumdar, R. Singh, M. Vatsa, “**Class Equilibrium Using Coulomb’s Law**”, *International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8, 2021.
10. **S. Chhabra**, G. Gupta, G. Gupta, M. Gupta “**Indian Currency Database for Forensic Investigations**”, In *Advances in Digital Forensics XVII: 17th IFIP WG 11.9 International Conference*, pp. 237-253, 2021.
11. Malhotra et al., “**Multi-Task Driven Explainable Diagnosis of COVID-19 using Chest X-ray Images**”, *Pattern Recognition*, 122, p.108243, 2022.
12. Mittal et al., “**A novel abnormality annotation database for covid-19 affected frontal Lung x-rays**”, *Plos One*, 17, no. 10, p. e0271931, 2022.
13. A. Singh, **S. Chhabra**, P. Majumdar, R. Singh, M. Vatsa, “**MANNET: A Large-Scale Duplicate Image Detection Dataset and Baseline Evaluations**”, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1780-1784, 2022.

Chapter 1

INTRODUCTION

Rapid advancements in machine learning (ML) and deep learning (DL) algorithms have significantly improved performance across a wide range of computer vision (CV) tasks, including face recognition, attribute prediction, object detection, image segmentation, and image classification. The high performance of deep neural networks (DNNs), particularly convolutional neural networks (CNNs), have driven the development of various artificial intelligence (AI)-based systems. These systems have found applications in diverse domains, such as autonomous vehicles, skin cancer diagnosis, targeted advertising, security and surveillance. Despite these successes, the applicability of AI systems in real-world scenarios remains a critical challenge. Two primary reasons underpin this limitation:

1. **Unethical Exploitation of AI Systems:** The growing misuse of AI technologies, including privacy violations and malicious applications, raises serious ethical and societal concerns (Hill, 2020), (Gnanasekar and Yanushkevich, 2019), (Pettit, 2017).
2. **Failures of Deep Models on Real-World Datasets:** Deep models often struggle to generalize effectively on real-world data due to issues such as data distribution shifts and lack of robustness during training (Intelligence, 2024), (University, 2022), (Geirhos *et al.*, 2020), (Oakden-Rayner *et al.*, 2020).

These challenges highlight the need for innovative solutions to ensure the responsible and effective deployment of AI systems in practical, real-world settings. The following subsections delve deeper into these issues and propose approaches to address them.

1.1 Unethical Exploitation of AI Systems

With the increasing applications of AI-based algorithms, the cases of unethical exploitation of these systems have increased significantly. Technological advancements enable criminals to perpetrate innovative crimes that are very difficult to prevent and detect.

One such example is face image profiling. A face is much more than just a person's identity. A profile picture on Instagram can reveal several protected and sensitive attributes like age, gender identity, race, sexual orientation, etc. Advances in AI algorithms and the growing trend of sharing images across the World Wide Web (including social media websites) have broadened the scope of information profiling and data analytics from photo collections. Low-cost cameras and advances in face analysis algorithms such as identity and attribute prediction make it easier for someone to take photos without permission and extract sensitive information like gender, age, identity, etc. by processing only their face image. Social Mapper (Wilkin, 2018a), FaceCheck (fac, 2023), and FindFace (Fin, 2019) are some of the apps where you need to upload your face photo to get the person's sensitive attribute information which can be misused for malicious purposes. There is also a growing trend of sharing images on social media platforms and CCTV cameras being used for surveillance. This poses a serious risk to an individual's privacy. A number of incidents (Gnanasekar and Yanushkevich, 2019) have highlighted the abuse of facial analytics for criminal identification and sexuality identification. For example, in a recent case from Michigan (Hill, 2020), an individual (later proven innocent) was arrested for a crime he didn't commit, only because a face-analysis algorithm made an incorrect prediction. These types of face analysis put the person in a difficult situation and caused panic, which can lead to depression and anxiety in extreme cases. The abuse of facial analytics also threatens the privacy of an individual who does not wish to be profiled, and leads to a loss of public trust.

To address these challenges, numerous countries have implemented data protection regulations, such as the Data Privacy Act (2012) (Commision, 2012) and the General Data Protection Regulation (GDPR) (GDP, 2018), which explicitly prohibit the unauthorized processing of personal data, including facial images. Compliance with such regulations requires systems that prevent unauthorized face-image analysis and allow users to selectively suppress or preserve attributes without compromising data utility.

Another concerning example of unethical exploitation is the generation of fake media. While AI advancements have brought significant benefits to fields like medicine, biometrics, and security, they also provide tools for malicious activities. The infamous DeepNude application (Kastrenakes, 2019), which used Deepfake technology to generate nude images of women, exemplifies such misuse. Although the app was eventually banned, its brief existence highlighted the darker side of Deepfake technology.

Other harmful uses include revenge porn, political manipulation, and the dissemination of false propaganda. With the increasing use of the internet and social media, Deepfake content poses significant risks to society by spreading misinformation and false narratives. The accessibility of these tools means that individuals with moderate computational resources can create convincing fake videos and images, threatening societal stability and democracy.

Researchers have developed algorithms to detect fake media, but these systems face challenges, especially with low-quality or highly compressed Deepfakes. Social media platforms apply varying compression techniques to shared media, further complicating detection efforts. For instance, (Kumar *et al.*, 2020) demonstrated that models trained on high-quality videos often perform poorly when tested on low-quality, compressed content. This performance gap arises due to:

1. Loss or alteration of artifacts learned during training on high-quality data.
2. Introduction of new artifacts due to compression.

Given these challenges, it is imperative to design robust fake media detection algorithms capable of handling diverse compression ratios and real-world conditions. Addressing these vulnerabilities will be crucial to mitigating the misuse of AI systems and protecting individual rights and societal trust.

1.2 Failure of AI Systems

Deep learning algorithms have become ubiquitous, powering a wide range of applications in areas such as computer vision, natural language understanding, autonomous systems, social media analytics, and AI for Social Good. Industries and government organizations are investing heavily in the development of sophisticated systems to address real-world challenges using deep learning. However, many of these systems fail to perform effectively in real-world scenarios. A recent study (Pandey, 2022) revealed that over 50% of AI systems underperform when deployed, resulting in significant monetary and time losses for organizations. The key reasons behind these failures include unreliable data and inherent biases in AI systems. Among the most critical factors are:

1. Lack of robustness and generalizability during training.

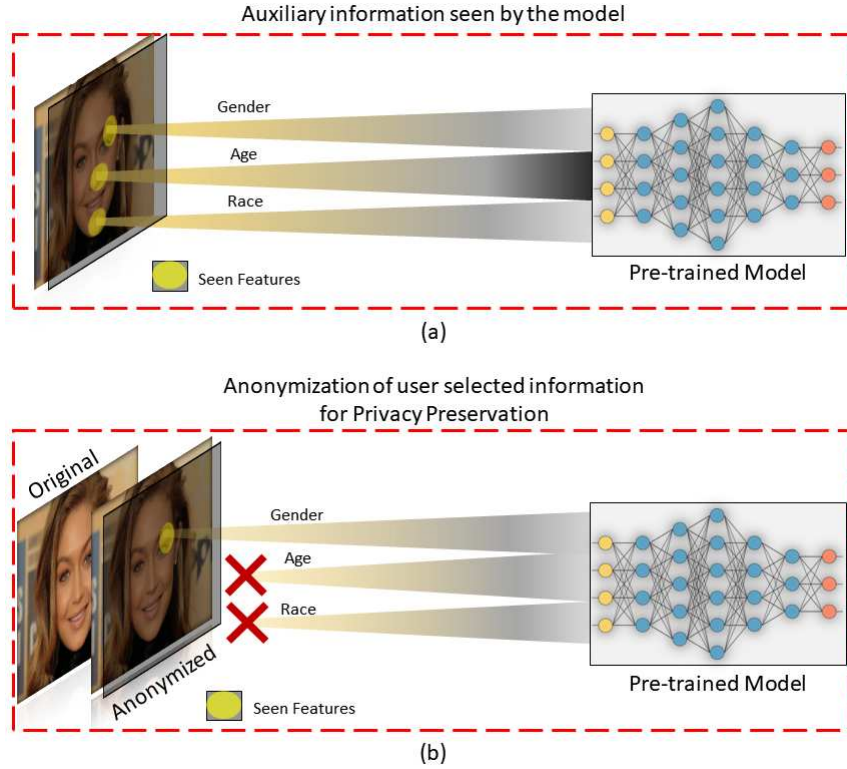


Figure 1.1: Illustrating the process of attribute prediction without and with the control mechanism. (a) shows the lack of control mechanism for attribute anonymization, and (b) shows the system with a control mechanism for attribute anonymization where the user can selectively suppress and preserve the attribute.

2. Data or distributional shifts during testing.

In supervised training settings, deep learning techniques excel at learning meaningful representations for classification tasks. These algorithms primarily aim to minimize loss with respect to given labels, optimizing performance on the training dataset. However, this generally leads to overfitting because of the reliance on the simplest patterns in the dataset, compromising generalizability. As a result, models perform well on benchmark datasets, but struggle with real-world data. To reconcile with this performance gap, recently, several researchers have explored this behaviour and attempted to understand the underlying reasons. In these researches, it has been found that the deep learning models frequently adopt shortcut strategies, focusing on dominant features that suffice for confident classification on the given dataset. This approach neglects other useful and distinct features that could enhance the model's robustness and adaptability. Consequently, the trained models lack of robustness and generalizability in diverse real-

world scenarios. In order to address this problem, it is important to learn the features ignored by the model along with the dominant feature to enhance the robustness and generalizability of the model in the real-world scenario.

Another major challenge is the shift in dataset characteristics between training and real-world testing. Changes in factors such as image quality, illumination, or pose variation during testing can degrade model performance. Since deep learning models heavily rely on the training dataset’s characteristics, such shifts can lead to significant performance drops. To mitigate the impact of data shifts, metric learning and transfer learning are commonly employed. These techniques refine the feature space by fine-tuning pre-trained models on the target dataset, thereby reducing intra-class variance and enhancing inter-class separability. However, fine-tuning involves updating millions of parameters, making it computationally expensive. Moreover, most commercial AI systems function as black boxes, with inaccessible model parameters, which limits the ability to fine-tune models for specific applications. Given these constraints, it is essential to develop solutions that address real-world data shifts without requiring extensive model retraining. Such approaches would not only enhance the practical utility of AI systems, but would also ensure their broader applicability across diverse real-world scenarios.

1.3 Focus of the Dissertation

To prevent the misuse of AI systems and enhance the robustness and generalizability of models in real-world scenarios, we investigated key challenges, including: 1) privacy breaches 2) fake media generation. 3) lack of robustness and generalizability in deep models and 4) data shifts in input, model, and feature spaces during training and testing. Our findings reveal that the root cause of these issues lies primarily in the input space.

Privacy breach: Advancements in face analysis algorithms have raised significant privacy concerns, particularly with non-consensual information profiling. Predicting sensitive attributes such as sexual orientation or being a drug addict without consent can cause distress and violate an individual’s privacy. As depicted in Figure 1.1(a), current face analysis systems lack mechanisms to regulate the attributes that model focuses on. This lack of control mechanism results in models revealing or predicting attributes indiscriminately, even when users have not provided explicit consent for such

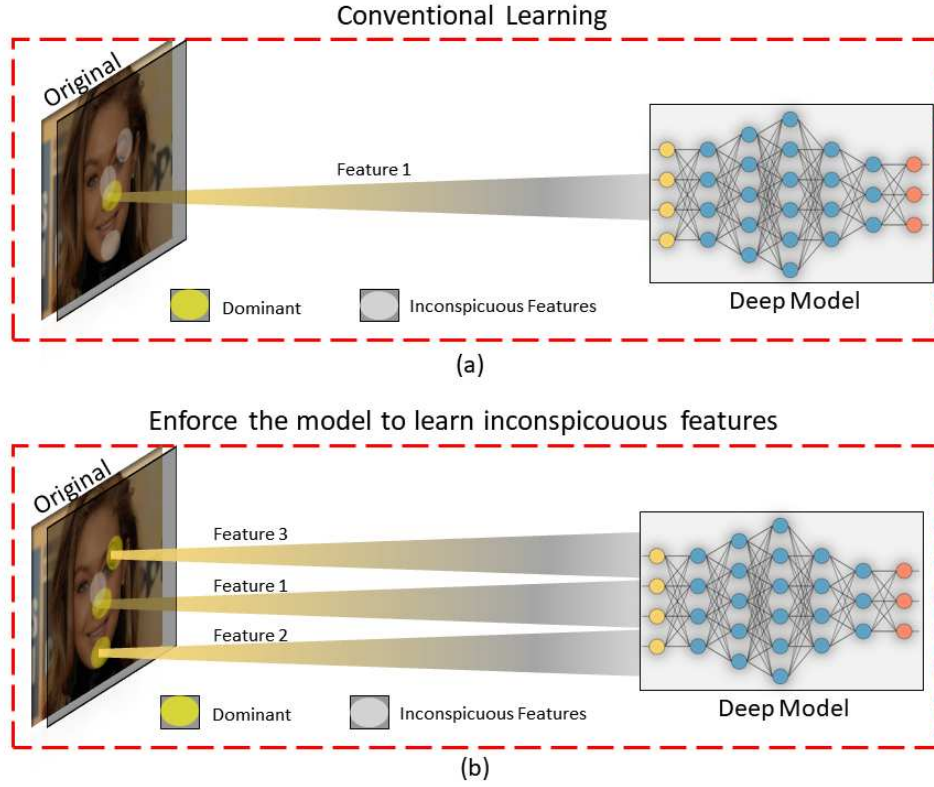


Figure 1.2: Illustrating the process of learning of the model. (a) shows the model ignored the relevant features and learned only dominant features (b) shows the model learned both dominant and inconspicuous features.

actions. Without a robust framework to manage this, individuals are left vulnerable to privacy breaches and unethical profiling. To address these challenges, we propose a novel solution that provides users with greater control over how their facial attributes are processed. Specifically, the proposed solution allows users to selectively anonymize and preserve facial attributes along with identity information. As illustrated in Figure 1.1(b), this solution introduces a mechanism to regulate the focus of deep learning models. By empowering users to determine which attributes are visible or obscured, the system not only protects individual privacy but also fosters trust in AI technologies by aligning their operations with ethical standards and privacy regulations. This approach has the potential to strike a balance between the utility of face analysis systems and the preservation of personal privacy.

Fake media generation: Platforms like WhatsApp, Twitter, and Instagram amplify the reach of such manipulated media, with millions of users sharing and consuming content daily. However, these platforms also introduce an additional layer of complex-

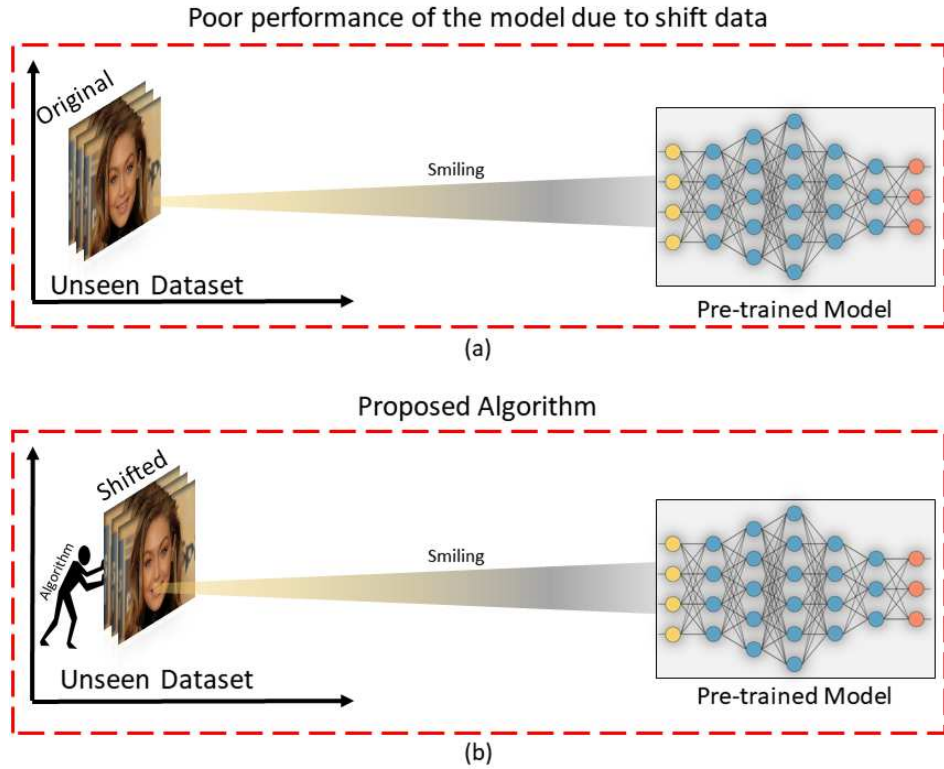


Figure 1.3: Illustrating the performance of the model due to data shift. (a) shows the poor performance of the model due to data shift. (b) illustrates the algorithm to shift the data without updating the model.

ity: implicit compression of media files. To optimize storage and speed up sharing, social media platforms automatically reduce the resolution and quality of uploaded images and videos. This compression process often strips away critical high-frequency details and artifacts that are typically leveraged by detection models to identify Deepfakes. As a result, traditional Deepfake detection methods, which are trained on high-quality datasets, struggle to perform effectively when faced with the low-quality, compressed media common on social platforms. This discrepancy between the training and real-world testing environments significantly hinders the ability to identify manipulated content, allowing Deepfakes to evade detection and propagate unchecked. To address these challenges, it is crucial to train robust Deepfake detection models specifically tailored for the low-quality media prevalent on social platforms.

Lack of robustness and generalizability of the deep model: Supervised learning algorithms aim to minimize classification loss to maximize accuracy. While these algorithms classify well on test data, the performance often deteriorates on out-of-

distribution (OOD) data samples. This occurs because models prioritize learning dominant features that are sufficient for classification, neglecting other discriminative yet less dominant features. For example, a facial recognition model that primarily focuses on the shape of the jawline may fail when the subject wears a mask or is partially occluded, as it has not learned to incorporate other facial features like eyes or forehead contours. As illustrated in Figure 1.2(a), a model trained in a controlled or benchmark environment prioritizes learning the most apparent or dominant feature patterns present in the data. external factors such as changes in lighting, pose, background, occlusion, and noise can significantly alter the appearance of the dominant features. In such cases, the model's over-reliance on these dominant features renders it ineffective, as the features it has learned may no longer be reliable. In real-world scenarios, external factors such as changes in lighting, pose, background, occlusion, and noise can significantly alter the appearance of the dominant features and limits the model's robustness against handling such variations. Addressing this issue requires models to learn both dominant and non-dominant features to improve robustness and generalizability as depicted in Figure 1.2(b).

Change in characteristics of the dataset during training and testing: In modern AI applications, commercial off-the-shelf (COTS) systems are widely employed for tasks such as facial recognition, emotion detection, and attribute prediction. However, their utility comes with a significant limitation i.e., they typically operate as black boxes, offering little to no insight into their internal workings or access to model parameters. This lack of transparency makes it challenging to adapt these systems to specific tasks or domains where the characteristics of the data may differ from those on which the models were originally trained. A critical challenge in using black-box models arises from data shifts, which are variations in input data between the training and deployment phases. These shifts can occur due to environmental differences, demographic diversity etc. Figure 1.3(a) illustrates how these shifts can lead to a significant deterioration in model performance. When confronted with data that deviates from its training distribution, the model may start focusing on incorrect or irrelevant features, resulting in inaccurate predictions. As shown in Figure 1.3(b), overcoming these challenges requires solutions that can adapt to data shifts without modifying the pre-trained models themselves.

1.4 Research Contribution

This dissertation focuses on developing innovative solutions to prevent the unethical exploitation of AI systems by addressing privacy breaches and the spread of fake media, while simultaneously improving the robustness and generalizability of models in real-world scenarios. The four major contributions of this dissertation are:

Attribute and Identity Anonymization for Privacy Preservation: A face image reveals a lot of ancillary information, which is used in a variety of applications. However, with the advancement in face analysis algorithm, extracting someones' facial attributes and identity becomes much easier. In order to prevent the information profiling, we proposed novel privacy preserving algorithm, which uses the concept of adversarial perturbation to suppress and preserve the selective facial attributes along with the identity without compromising the visual appearance of the image. The proposed algorithm addresses multiple challenges and several limitations of the existing for the privacy preservation algorithms including lack of control mechanism to selectively suppress and preserve attributes along with the identity and dependency of the algorithms on the reference image for the privacy preservation. Multiple experiments are performed on variety of databases with three different application scenarios. The results demonstrate that the proposed algorithm can handle multiple attribute anonymization process without affecting visual appearance and maintains a good balance between privacy and data utility.

Low Quality Deepfake Detection: Deepfakes are manipulated videos and audios that appear to be real, often created with the intention of spreading misinformation or propaganda. The Deepfakes are primarily distributed on social media platforms, which implicitly perform compression for faster transmission. In the literature, researchers have proposed various algorithms for the Deepfake detection. However, one of the major limitations of existing algorithms for Deepfake detection is failure on low quality Deepfake videos due to compression artifacts introduced while sharing it on social media platforms. In general, deep models learn features/artifacts in the data that may get suppressed during compression. It is possible to improve the performance of such models by learning from other (more) artifacts as well. These unseen imperceptible artifacts can be potentially helpful in the detection of low-quality Deepfakes, as they reduce the chances of losing relevant features during high compression. Therefore, we have

proposed an algorithm that enhances the unseen artifacts in the entire dataset and uses these unseen artifacts for further training the model. In order to suppress the already learned features and enhance the unseen imperceptible artifacts, we proposed to learn the visibility matrix. Experiments are performed on the multiple datasets to showcase the efficacy of the proposed algorithm.

Data Fine-tuning: Increasing demands of automated systems for face analysis has led to the development of commercial-off-the-shelf (COTS) systems. COTS systems act as a black box system due to the inaccessibility of the model parameters. In such scenarios enhancing the performance of black-box systems become a challenging task. To cope with the situation, a novel technique called data fine-tuning is proposed. Data fine-tuning refers to the process of adjusting the input data according to the behavior of the pre-trained model. Thus, a novel algorithm is proposed for data fine-tuning using adversarial perturbation. Multiple experiments are performed to evaluate the performance of the proposed algorithm. It is observed that data fine-tuning enhances the performance of black box models. A comparison of data fine-tuning with model fine-tuning is also performed. It is concluded that data fine-tuning outperforms model fine-tuning when the pre-trained model is trained on a large dataset due to its capability of better generalization. We also extended this algorithm to optimize the feature space of a pre-trained model for addressing the problem of data shift in real world scenarios. The proposed extended algorithm shifts the data in input space to obtain Feature Guided Perturbed (FGP) data by adding a uniform perturbation to the input data. Perturbation is learned by bringing the samples of the same class towards its class center and pushing away from other class centers in the feature space. Traditionally, perturbations are added to reduce the performance, whereas the proposed algorithm incorporates it for performance improvement. Multiple experiments are performed to evaluate the efficacy of the proposed algorithm for single and multiple attribute classification. It is observed that classes are better separated in the feature space by adding the learned uniform perturbation, thereby improving the overall classification performance.

DIVINE: Classical deep learning algorithms are generally known to focus on maximizing classification performance. This often leads to improper and insufficient learning, which does not generalize well when the model sees out-of-distribution samples. In this research, we propose a novel unified learning approach termed as *Diverse and Inconspicuous Learning* (DIVINE), which focuses on maximizing learning from a given

set of inputs and learning as many discriminative features as possible by computing dominance maps. We perform multiple experiments on several datasets including MNIST, CIFAR10, CIFAR100, TinyImageNet, and their corrupted and perturbed counterparts (CIFAR10-C, CIFAR10-P, CIFAR100-C, CIFAR100-P, TinyImageNet-C, and TinyImageNet-P), we demonstrate that DIVINE encourages the learning of a rich set of features. This, in turn, boosts the model’s robustness and its ability to generalize. The results on out-of-distribution datasets, such as those that are perturbed, achieve a performance 5.36%, 3.10%, and 21.85% mean Flip Rate (mFR) corresponding to CIFAR10-P, CIFAR100-P, and TinyImageNet-P datasets using DIVINE. On the other hand, Abridged Learning on CIFAR10-P, CIFAR100-P, and TinyImageNet-P datasets, achieve a performance 6.53%, 11.75%, and 31.90% mFR, respectively.

1.5 Organization of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 presents the first technical contribution, Anonymizing k-Facial Attributes via Adversarial Perturbations, which introduces a framework for selective anonymization through perturbation-based methods. Chapter 3 extends this idea with PrIdentity: Generalizable Privacy Preserving Adversarial Perturbations for Anonymizing Facial Identity, a more scalable and generalizable approach to identity protection. Chapter 4 shifts focus to manipulation detection, proposing Low Quality Deepfake Detection via Unseen Artifacts as a robust strategy for identifying forgeries generated by emerging pipelines. Chapter 5 explores Data Fine-tuning, a lightweight method for improving model robustness and generalization without retraining large models. Chapter 6 introduces Feature Guided Perturbation for Facial Attribute Classification, which leverages interpretable perturbations to enhance classification performance. Chapter 7 presents DIVINE: Diverse-Inconspicuous Feature Learning to Mitigate Abridge Learning, a framework for improving generalization by leveraging non-dominant features. Finally, Chapter 8 concludes the thesis by summarizing contributions, discussing ethical implications, and outlining promising directions for future research.

Chapter 2

Anonymizing k -Facial Attributes via Adversarial Perturbations

2.1 Introduction

“The face is the mirror of the mind, and eyes without speaking confess the secrets of the heart.” - Letter 54, St. Jerome.

Face analysis has been an area of interest for several decades in which researchers have attempted to answer important questions ranging from identity prediction (Zhou *et al.*, 2018), to emotion recognition (Li *et al.*, 2016), and attribute prediction (Abdulnabi *et al.*, 2015), (Sethi *et al.*, 2018). Focused research efforts and use of deep learning models have led to a high performance for tasks involved in face analysis. For instance, state-of-the-art results on YouTube Faces (Wolf *et al.*, 2011) and Point and Shoot Challenge (Beveridge *et al.*, 2013) databases are over 95%, and 97% (Goswami *et al.*, 2017), and attribute recognition on the Celeb-A database is more than 90% (Sethi *et al.*, 2018). While there are several advantages of these high accuracies, they also pose a threat to the privacy of individuals. As shown in Figure 2.1, several facial attributes such as age, gender, and race can be predicted from one’s profile or social media images. In a recent research, Wang and Kosinski predicted the “secrets of the heart”, such as predicting *sexual orientation* from face images (Wang and Kosinski, 2017). They reported 81% accuracy for differentiating between gay and heterosexual men with a single image and 74% accuracy is achieved for women. Similarly, targeted advertisements by predicting the gender and age from the profile and social media photographs have been a topic of research for the last few years.

Motivated by these observations, in this research, we raise an important question: “Can we anonymize certain attributes for privacy preservation and *fool* automated methods for attribute predictions?” The answer to this question relates to k -anonymity literature where information of each individual cannot be differentiated from the $k - 1$

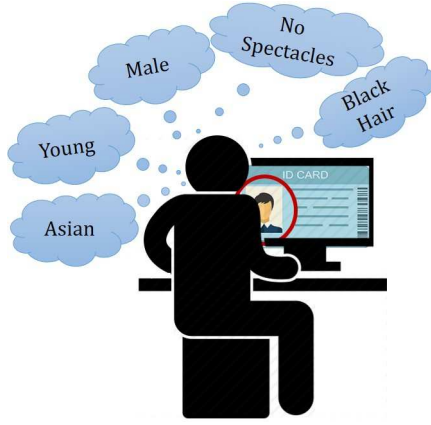


Figure 2.1: Profiling of a person using his face image in ID card for malicious purpose.

Table 2.1: Summarizing the attribute suppression/anonymization algorithms in the literature.

Author	Method	No. of Attributes	Dataset	Controlling Attributes	Visual Appearance
Othman and Ross, 2014	Face Morphing and fusion	One	MUCT	No	Partially Preserved
Mirjalili and Ross, 2017	Delaunay Triangulation and fusion	One	MUCT, LFW	No	Partially Preserved
Mirjalili <i>et al.</i> , 2017	Fusion using Convolutional Autoencoder	One	MUCT, LFW, CelebA, AR-Face	No	Preserved
Rozsa <i>et al.</i> , 2016, 2017	Fast Flipping Attribute	Multiple	CelebA	No	Preserved
Mirjalili <i>et al.</i> , 2020	PrivacyNet (GAN based model)	Multiple	MUCT, MORPH, RaFD, CelebA, UTK-face	Yes	Partially Preserved
Li <i>et al.</i> , 2021	Anonymization using StarGAN	Multiple	VGGFace2, CelebA	Yes	Partially Preserved
Li <i>et al.</i> , 2023	Semantic Aware Method	Multiple	VGGFace2-HQ, CelebA HQ	No	Partially Preserved
Rot <i>et al.</i> , 2024	Fusion of disentangled spatial segments	One	1000 male images, 1000 female images	No	Partially Preserved
Proposed	Adversarial Perturbations	Multiple	MUCT, LFWcrop, Celeb-A	Yes	Preserved

individuals (Sweeney, 2002). It differs from the fact that in attribute anonymization, certain attributes are to be suppressed/anonymized while remaining ones are retained. For instance, if the images are uploaded to the driving license database to ascertain identity, it should not be used for any other facial analysis, except identity matching.

2.1.1 Literature Review

In the literature, privacy preservation in face images has been studied from two perspectives. Researchers have studied this problem in terms of privacy-preserving biometrics while others have termed it as attribute suppression. As mentioned above, face images reveal a lot of ancillary information for which the user may not have consented. In order to protect such ancillary information (soft biometrics), researchers have proposed

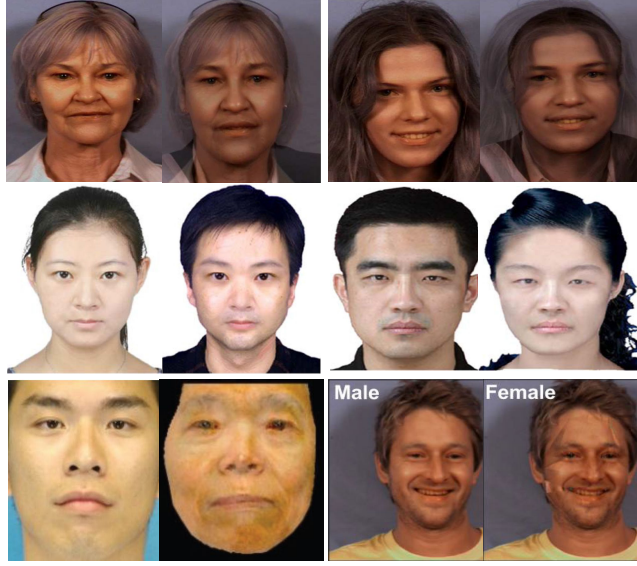


Figure 2.2: Demonstrating the attribute anonymized images generated by existing algorithms. First and third column images are original images while second and fourth column images are gender anonymized images (Sim and Zhang (2015), Suo *et al.* (2011), Othman and Ross (2014), Mirjalili and Ross (2017)).

several different methodologies. Boyle *et al.* (2000) developed an algorithm to blur and pixelate the images in the video. Newton *et al.* (2005) have developed an algorithm for face de-identification in video surveillance such that the face recognition fails while preserving other facial details. Gross *et al.* (2006) have shown that distorting image via blurring and pixelation method gives poor results. To improve the results, they proposed model based face de-identification method for privacy protection. Some researchers have also worked on the privacy of soft biometrics. Othman and Ross (2014) have proposed attribute privacy preserving technique, in which the soft biometrics attribute such as gender is “flipped” while preserving the identity for face recognition. In order to flip the gender, face morphing scheme is used in which the face of other opposite gender is morphed with the input image. As an extension of this work, Mirjalili and Ross (2017) and Mirjalili *et al.* (2017) have proposed Delaunay triangulation, and convolutional autoencoders based methods to flip gender information while preserving face identity. Suo *et al.* (2011) have presented an image fusion framework in which the template of opposite gender face image is taken for fusion with the candidate image while preserving face identity. Jourabloo *et al.* (2015) have developed an algorithm for de-identification of face image while preserving other attributes. For attribute preservation, it uses k images (motivated by k -Same) which shares the same attributes for fusion. Sim and Zhang (2015) have proposed a method which independently controls

the identity alteration and preserves the other facial attributes. It decomposes the facial attribute information into different subspaces to independently control these attributes.

To anonymize the facial attributes, Rozsa *et al.* (2017) have proposed deep learning based model for facial attribute prediction. A deep convolutional neural network is trained for each attribute separately and in order to test the robustness of the trained model, adversarial images are generated using fast flipping attribute (FFA) technique. As an extension of their work Rozsa *et al.* (2016) have used FFA and adversarial images are generated in which a facial attribute is flipped. They have observed that the few attributes are effected while flipping the targeted attribute. For instance, while changing the ‘wearing lipstick’ attribute, other attributes such as ‘attractive’ and ‘heavy makeup’ are also flipped. Mirjalili *et al.* (2020) introduced a semi-adversarial network (SAN) framework designed to obfuscate multiple facial attributes while preserving essential image quality and usability. Shen *et al.* (2019) introduced a human-sensitivity-aware image perturbation model that protects sensitive attributes from machine recognition while remaining imperceptible to humans. It is the first to propose a human sensitivity map, derived from human studies, to guide privacy-preserving modifications in images. Li *et al.* (2021) proposed an adaptive method for facial attribute obfuscation, preserving key identity-independent features while anonymizing sensitive attributes. Li *et al.* (2023b) proposed a semantic-aware approach to face obfuscation using attribution maps, which identify salient facial regions contributing to identity recognition. The method utilizes deep learning models to generate obfuscation masks guided by these attribution maps, selectively altering identity-sensitive features while preserving non-sensitive details. Rana *et al.* (2023) proposed a novel privacy protection method that adds intelligent, attribute-specific noise to images using FGSM and GradCam++, ensuring sensitive attributes are perturbed while preserving overall aesthetics. To address the challenge of gender privacy, Rot *et al.* (2024) work proposes a novel method that fuses disentangled spatial segments of facial images. By decomposing facial features into independent components and selectively fusing them into a reconstructed image, the approach ensures that gender-related information is obscured while retaining other non-sensitive details. Figure 2.2 shows some sample attribute anonymized images from existing literature.

Based on the literature review, we observe that there are algorithms for single attribute anonymization as summarized in Table 2.1. However, there are three major

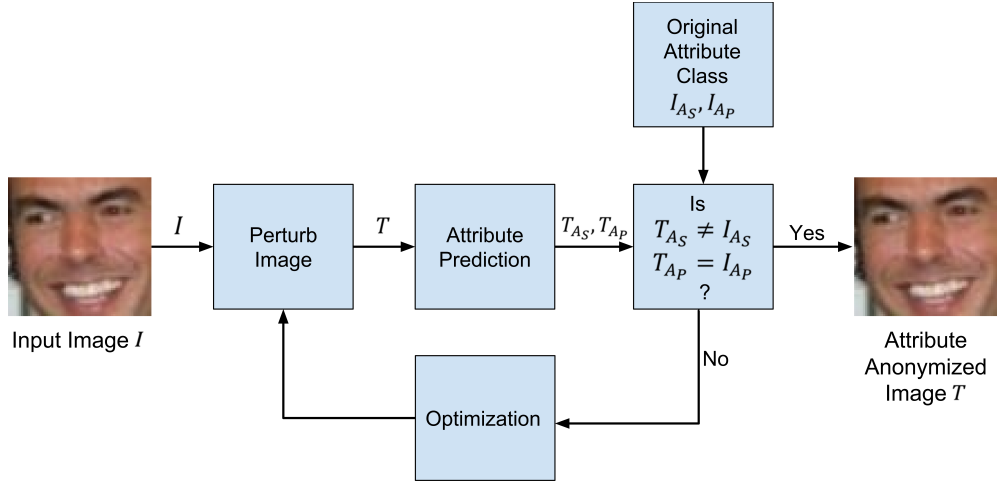


Figure 2.3: Illustrating the steps involved in the proposed algorithm.

challenges in anonymizing multiple attributes.

1. While anonymizing facial attributes, there should be no visual difference between original and anonymized images.
2. Selectively anonymizing few and retaining some attributes require a “control” mechanism. For example, gender and expression can be anonymized while retaining race and eye color and other attributes such as attractiveness and hair color may be in “do not care” condition.
3. In applications involving matching faces for recognition, identity should be preserved while anonymizing attributes.

The major limitation of these methods is that they do not address the first two challenges mentioned above. Existing algorithms primarily depend on a candidate image which is fused with the original image. This privacy preserving fusion leads to major transform and loss in visual appearance. Rozsa *et al.* (2016) have addressed this issue but due to lack of any control mechanism, the other attributes are also suppressed, while suppressing one attribute.

2.1.2 Research Contributions

This research proposes a novel algorithm for privacy-preserving k -attribute anonymization using adversarial perturbation. The proposed algorithm jointly anonymizes and preserves multiple facial attributes without affecting the visual appearance of the image. The proposed algorithm is also extended for identity preserving k -attributes anonymization. Experiments on three databases and comparison with existing techniques showcase the efficacy of the proposed algorithm.

2.2 Proposed Approach

The problem statement can formally be defined as: “create an image such that a set of pre-defined attributes are preserved while another set of pre-defined attributes are suppressed”. As shown in Figure 2.3, the proposed algorithm can change the prediction output of certain attributes from the true class to a different target class. The detailed description of the proposed algorithm is given below:

Let $\mathbf{I}$ be the original face image with k number of attributes in the attribute set $\mathbf{A}$. For each attribute $\mathbf{A}_i$, there are C_i number of classes. For instance, in attribute ‘Gender’, two classes are {Male, Female} and ‘Expression’ attribute has five classes, namely {Happy, Sad, Smiling, Anger, Fearful}. Mathematically, it is written as:

$$\mathbf{A} = \{\mathbf{A}_1(C_1), \mathbf{A}_2(C_2), \dots, \mathbf{A}_k(C_k)\} \quad (2.1)$$

Let $\mathbf{T}$ be the attribute anonymized image generated by adding perturbation $\mathbf{w}$ in the original image $\mathbf{I}$ where range of $\mathbf{T}$ is between 0 to 1. Mathematically, it is written as:

$$\mathbf{T} = \mathbf{I} + \mathbf{w} \quad (2.2)$$

$$\text{such that } \mathbf{T} \in [0, 1]$$

To satisfy the above constraint, $\tanh$ function is applied on $\mathbf{I} + \mathbf{w}$ as follows:

$$\mathbf{T} = \frac{1}{2}(\tanh(\mathbf{I} + \mathbf{w}) + 1) \quad (2.3)$$

In an attribute suppressing/preserving application, let $\mathbf{A}_S$ and $\mathbf{A}_P$ be the sets of attributes to be suppressed and preserved where, $\mathbf{A}$ is the superset of $(\mathbf{A}_S \cup \mathbf{A}_P)$ and $\mathbf{A}_S \cap \mathbf{A}_P = \phi$. In k - attribute anonymization task, it has to be ensured that the class of $\mathbf{A}_S$ attributes in $\mathbf{T}$ changes to some other class while preserving $\mathbf{A}_P$ attributes in image $\mathbf{T}$.

$$\mathbf{T}_{\mathbf{A}_S} \neq \mathbf{I}_{\mathbf{A}_S}, \mathbf{T}_{\mathbf{A}_P} = \mathbf{I}_{\mathbf{A}_P} \quad (2.4)$$

In order to suppress and preserve the sets of attributes $\mathbf{A}_S$ and $\mathbf{A}_P$ respectively, the distance between attributes of $\mathbf{A}_P$ in $\mathbf{I}$ and $\mathbf{T}$ is minimized, while the distance between attributes $\mathbf{A}_S$ is maximized. The objective function is thus represented as:

$$\min [D(\mathbf{I}_{\mathbf{A}_P}, \mathbf{T}_{\mathbf{A}_P}) - D(\mathbf{I}_{\mathbf{A}_S}, \mathbf{T}_{\mathbf{A}_S})] \quad (2.5)$$

$$\text{such that } \mathbf{T}_{\mathbf{A}_S} \neq \mathbf{I}_{\mathbf{A}_S}, \mathbf{T}_{\mathbf{A}_P} = \mathbf{I}_{\mathbf{A}_P}$$

where, D is the distance metric. To preserve the visual appearance of the image, the distance between $\mathbf{I}$ and $\mathbf{T}$ is also minimized. Experimentally, we found that the ℓ_2 distance metric is most suitable for preserving the visual appearance of the image. Equation 2.5 is updated as:

$$\min \{D(\mathbf{I}_{\mathbf{A}_P}, \mathbf{T}_{\mathbf{A}_P}) - D(\mathbf{I}_{\mathbf{A}_S}, \mathbf{T}_{\mathbf{A}_S}) + \|\mathbf{I} - \mathbf{T}\|_2^2\} \quad (2.6)$$

$$\text{such that } \mathbf{T}_{\mathbf{A}_S} \neq \mathbf{I}_{\mathbf{A}_S}, \mathbf{T}_{\mathbf{A}_P} = \mathbf{I}_{\mathbf{A}_P}$$

The term $[D(\mathbf{I}_{\mathbf{A}_P}, \mathbf{T}_{\mathbf{A}_P}) - D(\mathbf{I}_{\mathbf{A}_S}, \mathbf{T}_{\mathbf{A}_S})]$ in the above Equation and the first two constraints i.e. $\mathbf{T}_{\mathbf{A}_S} \neq \mathbf{I}_{\mathbf{A}_S}, \mathbf{T}_{\mathbf{A}_P} = \mathbf{I}_{\mathbf{A}_P}$ are non linear. Therefore, to solve the same, an alternative function $f(\mathbf{T})$, inspired from Carlini and Wagner (2017), is used. The above Equation is thus written as:

$$\min \{f(\mathbf{T}) + \|\mathbf{I} - \mathbf{T}\|_2^2\} \quad (2.7)$$

Here, $f(\mathbf{T})$ attempts to preserve the attributes $\mathbf{A}_P$ and suppress attributes $\mathbf{A}_S$. There can be multiple cases for facial attribute anonymization, the objective function for each case is discussed as follows:

Case I - Single Attribute Anonymization: This case formulates the scenario pertaining to changing the class of a single attribute i.e., the set $\mathbf{A}_S$ contains only one attribute, $\mathbf{U}$. In order to change the class of a single attribute $\mathbf{U}$ with class i to any other class j where $(j \neq i)$, the objective function $f(\mathbf{T})$ is formulated as:

$$f(\mathbf{T}) = \max \{0, \max(P(\mathbf{U}|\mathbf{T})) - P(U^j|\mathbf{T})\} \quad (2.8)$$

where, $P(x)$ denotes a function giving the probability value of x . For our case, we have used the Softmax output score (discussed in Section 4). The term $\max(P(\mathbf{U}|\mathbf{T}))$ used in Equation 2.8 outputs the maximum class score of attribute $\mathbf{U}$ and term $P(U^j|\mathbf{T})$ outputs the score of each class of attribute $\mathbf{U}$ except the i^{th} class. It is important to note that Equation 2.8 can also provide the target attribute class by giving a class label to the variable j .

Case II - Multiple Attribute Preservation and Suppression: This case formulates the scenario when the prediction output of multiple attributes, e.g. ‘Gender’ and ‘Attractiveness’ are suppressed while preserving other attributes e.g. ‘Ethnicity’, ‘Wearing glasses’, and ‘Heavy makeup’. In this specific example, the set $\mathbf{A_P}$ contains a list of three attributes whereas, $\mathbf{A_S}$ contains list of two attributes. For such scenarios of multiple attributes, the function $f(\mathbf{T})$ will be the summation of Equation 2.8 for each attribute. Mathematically, it is expressed as:

$$f(\mathbf{T}) = \sum_{\mathbf{U} \in \{\mathbf{A_S}, \mathbf{A_P}\}} \max\{0, \max(P(\mathbf{U}|\mathbf{T})) - P(U^j|\mathbf{T})\} \quad (2.9)$$

In order to preserve the attributes, the target attribute class will be same as input attribute class. The optimization method in Equation 2.9 can control the separation between target class score and next maximum class score by providing value ‘ $-c$ ’ in place of 0 where $c \in [0, 1]$, i.e.

$$f(\mathbf{T}) = \sum_{\mathbf{U} \in \{\mathbf{A_S}, \mathbf{A_P}\}} \max\{-c, \max(P(\mathbf{U}|\mathbf{T})) - P(U^j|\mathbf{T})\} \quad (2.10)$$

Case III - Identity Preservation: The third and the most relevant case with respect to face recognition applications, is preserving the identity $\mathbf{Id}$ of a person while suppressing an attribute. The objective function in this case can be written as:

$$\min \{f(\mathbf{T}) + \|\mathbf{I} - \mathbf{T}\|_2^2 + \mathcal{D}(\mathbf{Id_I}, \mathbf{Id_T})\} \quad (2.11)$$

Here, $\mathbf{Id_I}$ and $\mathbf{Id_T}$ is the identity of the face images $\mathbf{I}$, and $\mathbf{T}$ respectively obtained from any face recognition algorithm, and $\mathcal{D}$ is the distance metric to match two face

features.

The proposed formulation of k -anonymizing attributes can be viewed as adding adversarial noise such that some attributes are suppressed while preserving selected attributes and identity information. Figure 2.4 shows some examples of attribute anonymization using the proposed algorithm.

2.3 Datasets and Experimental Details

The proposed algorithm is evaluated on three datasets: MUCT (Milborrow *et al.*, 2010a), LFWcrop (Huang *et al.*, 2007), and CelebA (Liu *et al.*, 2015a). Comparison has been performed with the algorithm proposed by Mirjalili and Ross (2017). The details of the databases are described as follows:

MUCT dataset contains 3,755 images of 276 subjects out of which 131 are male and 146 are female captured under varying illuminations using five webcams. Viola-Jones (Viola *et al.*, 2005) face detector is applied to all images and the detector fails to detect 49 face images. Therefore, only 3,706 images are used for further processing. **LFWcrop dataset** contains 13,233 face images of 5,749 subjects. In order to evaluate the performance, view 2 of the dataset which consist of 6,000 pairs of images, has been considered. Out of these 6,000 pairs of images, only one image from each pair has been selected for anonymization. **CelebA dataset** contains 202,599 face images of celebrities. From this dataset, only the test set of 19,962 images has been considered for attribute anonymization. The results of anonymizing a single attribute are shown on all three databases while the results of anonymizing multiple attributes are shown on CelebA dataset.

As shown in Table 2.2, three experiments are performed, one corresponding to each case discussed in Section 2.2. For attribute anonymization task, white box attacks are performed whereas for identity preservation, black box attack is performed. The attributes are anonymized corresponding to the attribute classification model while the identities are preserved according to face recognition algorithm. For attribute classification, we have selected fine-tuned VGGFace (Parkhi *et al.*, 2015) and for identity preservation, OpenFace (Amos *et al.*, 2016) model is used. The distance metrics used with both the models is Euclidean distance. The performance of face recognition is eval-

Table 2.2: The experiments are performed pertaining to three cases to showcase the effectiveness of the proposed algorithm.

Experiment	Dataset	# Attributes Anonymized	Attribute Anonymized	
			Suppressed	Preserved
Single Attribute	MUCT, Celeb-A, LFWcrop	1	Gender	-
Multiple Attributes	Celeb-A	3, 5	Gender, Attractive, Smiling	Heavy makeup, High cheekbones
Identity Preservation	MUCT, LFWcrop	1+1	Gender	Identity

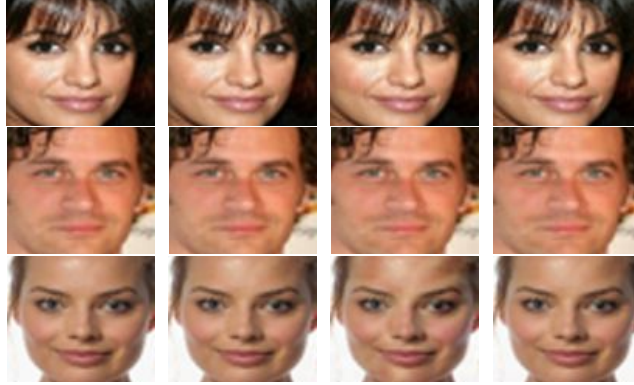


Figure 2.4: This figure shows the images before and after anonymizing attributes of Celeb-A dataset. First column represents the original images while second, third, and fourth columns show one, three, and five attributes anonymized images.

uated on both OpenFace (Amos *et al.*, 2016) and VGGFace (black box model) (Parkhi *et al.*, 2015).

Implementation details: The proposed algorithm is implemented in Tensorflow with 1080 Ti GPU. For learning the perturbation, L2 attack has been performed with Adam optimizer. The learning rate is set 0.01 and number of iterations used is 10000.

Table 2.3: Confusion matrix displaying classification accuracies (%) of before and after suppression of three attributes together on the CelebA dataset

		Attribute Class	Prediction		Attribute Class	Prediction		Attribute Class	Prediction	
			Male	Not Male		Smiling	Not Smiling		Attractive	Not Attractive
Ground Truth	Before	Male	87.70	12.30	Smiling	64.59	35.41	Attractive	89.31	10.69
		Not Male	19.64	80.36	Not Smiling	24.66	75.34	Not Attractive	28.41	71.59
	After	Male	3.89	96.11	Smiling	0.02	99.98	Attractive	0.28	99.72
		Not Male	100	0	Not Smiling	99.90	0.10	Not Attractive	99.59	0.41

2.4 Performance Evaluation

The proposed model is evaluated for single and multiple attribute anonymization. For multiple attribute anonymization, two types of experiments are performed: attribute suppression, and attribute suppression + preservation. In case of single attribute anonymization, the performance of the proposed algorithm has been analyzed for the tasks of attribute suppression as well as identity preservation. To generate the attribute anonymized images, only those samples have been considered which are correctly classified by the attribute classification model.

2.4.1 Multiple Attribute Anonymization

The proposed algorithm for multiple attribute anonymization is evaluated on the CelebA dataset. Three attributes are considered for the task of attribute suppression, while attribute suppression and preservation is performed with five attributes.

Attribute Suppression: Experiments have been performed with three attributes i.e. ‘Gender’, ‘Smiling’, ‘Attractive’. The confusion matrix in Table 2.3 shows that in over 96% of the images, attributes are correctly suppressed. After applying the proposed algorithm for the attribute ‘Attractive’, the classification accuracy dropped from 89.31% to 0.28%; thereby showcasing the efficacy of the proposed technique for the task of attribute suppression.

Attribute Suppression and Preservation: The results for five attributes ‘Gender’, ‘Attractive’, ‘Smiling’, ‘Heavy Makeup’ and ‘High Cheekbones’ are shown in Figure 2.5. Three attributes are suppressed i.e. ‘Gender’, ‘Attractive’ and ‘Smiling’, while remaining two are preserved. The confidence value (‘c’ value as defined in Equation 2.10) is set to 0.1. In Figure 2.5, the first and second row histograms show the attribute class score distributions before and after anonymization of images. It is observed that the attribute class score distributions of first three attributes ‘Gender’, ‘Attractive’ and ‘Smiling’ are flipped while the class score distributions of ‘Heavy Makeup’ and ‘High Cheekbones’ are preserved. This illustrates the utility of the proposed algorithm for multiple attributes suppression and preservation. Figure 2.4 presents the original and

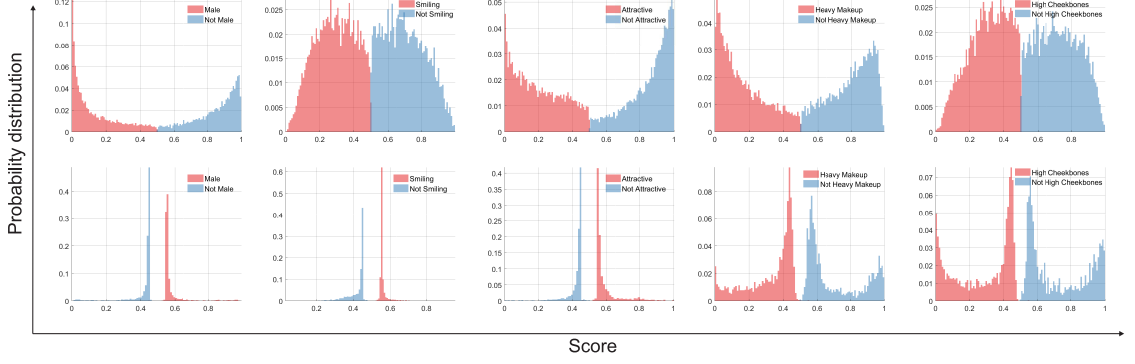


Figure 2.5: Comparing the attribute class score distributions of images before and after anonymization of 5 attributes together on the CelebA dataset. The first row distributions pertain to the original images whereas, second row distributions corresponds to the anonymized images.

anonymized images for one, three, and five attributes. The similarity of visual appearance between original and modified images further strengthens the usage of the proposed algorithm.

2.4.2 Attribute Suppression with Identity Preservation

To evaluate the performance of single attribute suppression with identity preservation, experiments are performed on MUCT, CelebA and LFWcrop datasets. The results of single attribute suppression and attribute suppression along with identity preservation are discussed below.

Single Attribute Suppression: Figure 2.6 shows the result of the ‘Gender’ attribute suppression on CelebA dataset. It can be observed that the score distribution of attributes are completely flipped before and after anonymization. The confusion matrix

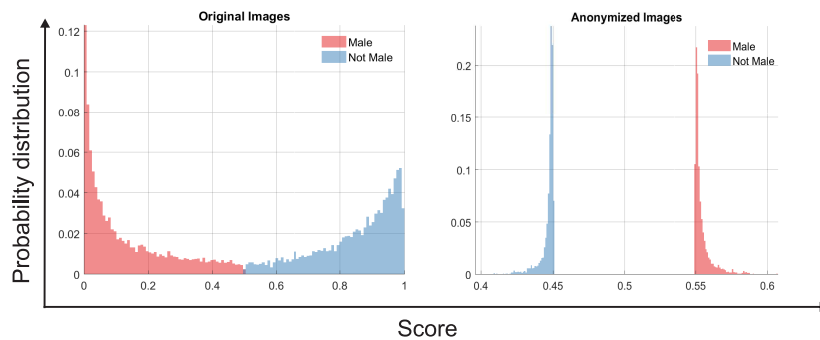


Figure 2.6: Comparing score distributions before and after anonymizing ‘Gender’ attribute on the CelebA dataset.

Table 2.4: Comparison of confusion matrix of the proposed algorithm with Mirjalili and Ross (2017) on the MUCT dataset.

Anonymization	Attribute Class	Mirjalili and Ross (2017)		Proposed	
		Male	Female	Male	Female
Before	Male	1762	17	1741	87
	Female	521	1300	252	1626
After	Male	276	1503	0	1828
	Female	1255	566	1878	0



Figure 2.7: Examples of before and after anonymization of ‘Gender’ attribute on MUCT and LFWcrop datasets. First row images are original images, second row images are ‘gender’ anonymized images.

for MUCT dataset before and after anonymization is shown in Table 2.4. On comparing the results with the algorithm proposed by Mirjalili and Ross (2017), it can be observed that after anonymization, all samples are misclassified by the proposed method. It is important to note that the method proposed by Mirjalili and Ross (2017) is dependent on the fusion of the other candidate images which embeds visual distortion in the image. On the other hand, in the proposed method, the anonymization of attribute is independent of another candidate image, thus resulting in limited visual distortions. Some example images of a suppressed attribute on MUCT and LFWcrop datasets are demonstrated in Figure 2.7. These samples show that there are minimal effect on visual appearance of the image after attribute anonymization.

Identity Preservation: To evaluate identity preservation performance, while anonymizing an attribute, experiments are performed on both MUCT and LFWcrop datasets. For MUCT dataset, a single image gallery and two probe sets are used. The first probe set consist of original images while the second probe set contains the corresponding ‘Gender’ suppressed images of probe set 1. Two widely use face recognition models i.e. OpenFace (Amos *et al.*, 2016) and VGGFace (Parkhi *et al.*, 2015) are used for (i) original to original face matching (i.e., gallery to probe set 1) and (ii) original to anonymized face matching (i.e., gallery to probe set 2). The Cumulative Match Char-

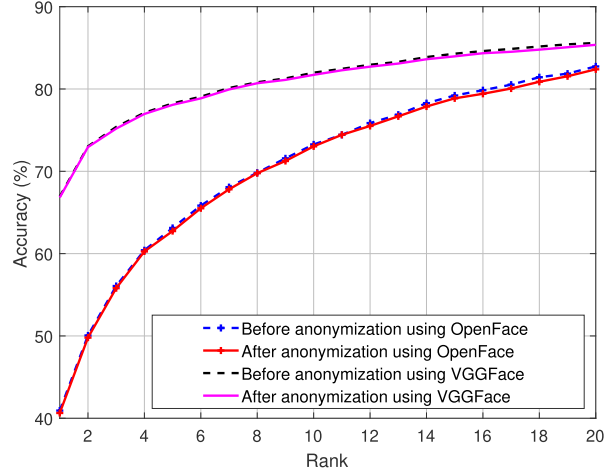


Figure 2.8: CMC curve showing original to original face matching vs original to anonymized face matching on the MUCT dataset.

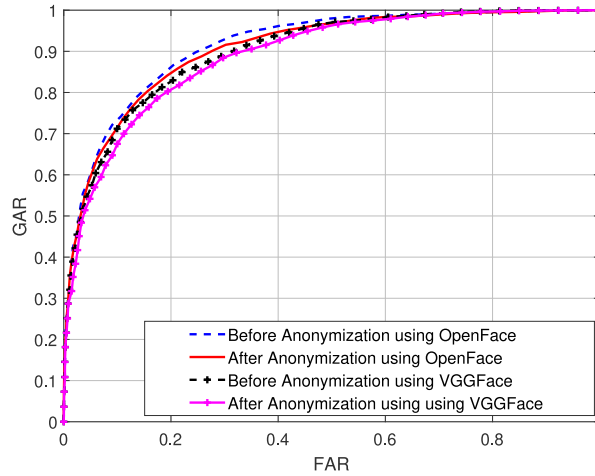


Figure 2.9: ROC curve showing face recognition performance on the LFWcrop Dataset

acteristic (CMC) curves summarizing the performance of face recognition before and after suppressing the attributes are show in Figure 2.8. The low variation observed in accuracy for recognition before and after anonymization motivates the applicability of the proposed algorithm for identity preservation as well.

Figure 2.9 shows the Receiver Operating Characteristic (ROC) curves for face verification on LFWcrop dataset. Experiments are performed on (i) 6000 original pair images, and (ii) 6000 original-anonymized pairs. Face verification is performed using OpenFace (Amos *et al.*, 2016) and VGGFace model (Parkhi *et al.*, 2015). The results show minimal effect on the face verification performance before and after attribute anonymization.

2.5 Robustness Against Adversarial Defenses

While the proposed algorithm effectively learns adversarial perturbations for attribute anonymization in a white-box setting, its robustness against adversarial defenses has not been explored. In practice, the attribute classifiers or downstream systems may employ various adversarial defense mechanisms such as adversarial training, input denoising, purification, or randomized smoothing that could potentially mitigate or nullify the anonymization effect. Evaluating the perturbation’s effectiveness under such defended environments, and developing defense-aware or adaptive anonymization strategies, constitute important directions for future work.

2.6 Summary

Attribute anonymization while preserving identity has several privacy preserving applications. This research presents a novel algorithm based on adversarial noise addition concept such that selected attributes (or features) are anonymized and selected attributes (including identity information) are preserved for automated processing. Experiments are performed on CelebA, LFWcrop and MUCT databases with three different application scenarios. The results demonstrate that the proposed algorithm can handle multiple attribute anonymization process without affecting visual appearance and face recognition performance.

Chapter 3

PrIdentity: Generalizable Privacy Preserving Adversarial Perturbations for Anonymizing Facial Identity

3.1 Introduction

Sharing images on social media platforms such as Facebook, Twitter, and Instagram have increased significantly. As per Global Social Media Stats 2024 (dig, 2024), the number of social media users from January 2023 to January 2024 increased by 320 million worldwide. However, with the rise in the number of social media users, identity theft (ide, 2024), (ide, 2023) and financial fraud (Kapoor, 2024), (Alam, 2023)) via online social media profiling have also increased. With the advancement in machine learning and deep learning algorithms, accessing someone's identity information from face images has become an easy task. Social Mapper (Wilkin, 2018b), FaceCheck (fac, 2023), and FindFace (Fin, 2019)) are a few applications where users need to upload a person's face image to obtain that person's identity information, which can be further misused for malicious purposes. Figure 3.1 illustrates the information profiling via social media platforms. This leads to a decline in public trust in social media and presents a serious threat to an individual's privacy.

It is essential to anonymize facial identity information in images to minimize misuse due to automatic identity leakage. Several researchers have proposed algorithms based on k-anonymity (Samarati and Sweeney, 1998), (Gross *et al.*, 2005), morphing (Korshunov and Ebrahimi, 2013), Korshunov and Ebrahimi (2014), generative modeling (Chen *et al.*, 2021c), (Pan *et al.*, 2021), (Laishram *et al.*, 2024), (Thakral *et al.*, 2024), and Deepfakes (Zhu *et al.*, 2020), (Cai *et al.*, 2024) that synthesize/fuse the input face image with the target image for identity anonymization. However, fusing input with the target face image distorts the input image's visual appearance and hinders its applicability in real-world scenarios. To address this problem, it is essential to design an

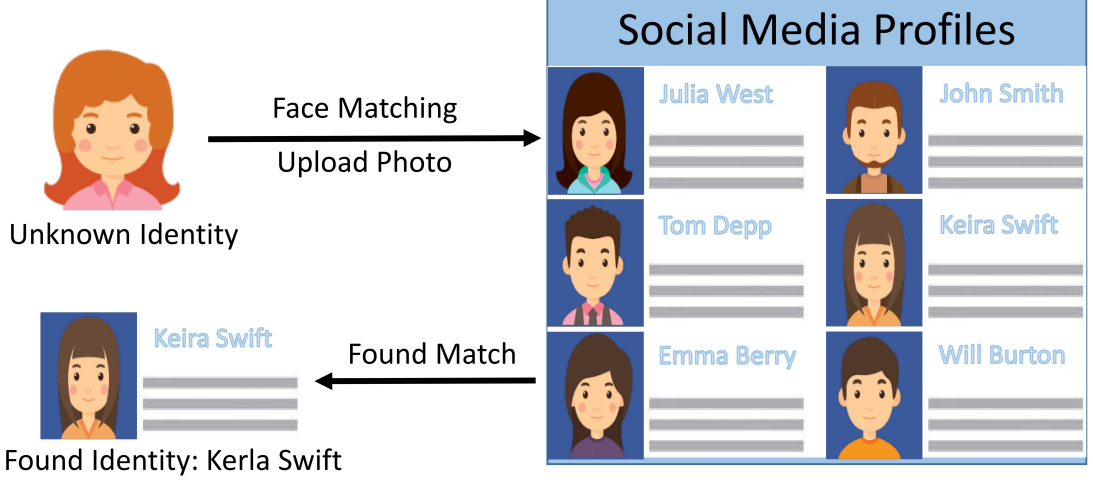


Figure 3.1: Illustration of information profiling of unknown identity from social media platforms.

algorithm with three basic properties: 1) **Privacy**: The algorithm should modify the input face image such that the Identity Information is suppressed. 2) **Data Utility**: There should be minimal difference between the visual appearance of the original and modified image. 3) **Generalizability**: The algorithm should perform well across different unseen face-recognition models.

In this research, we have proposed a novel **Privacy Preserving Identity Anonymization (PrIdentity)** algorithm that utilizes the concept of adversarial perturbations and satisfies the three properties to anonymize the identities of input face images. To ensure privacy and data utility, the proposed algorithm learns a perturbation such that the distance between the face representation of the original and the perturbed image is maximized while preserving the visual appearance of the perturbed image. In order to maintain a good balance between data utility and privacy, it is important to use a regularization technique for learning the adversarial perturbation. It has been shown in the literature that $\mathcal{L}_1$ and $\mathcal{L}_2$ norms are well-adopted and commonly used regularization techniques, where $\mathcal{L}_1$ norm introduces sparsity and $\mathcal{L}_2$ norm imposes convexity to reduce overfitting. However, using either $\mathcal{L}_1$ norm or $\mathcal{L}_2$ norm for learning adversarial perturbation could result in overfitting towards privacy or data utility and limit the applicability of the proposed PrIdentity algorithm. To address this problem, we have proposed to learn a parameter p in $\mathcal{L}_p$ norm, which introduces the adequate amount of sparsity and convexity to avoid overfitting and maintains a good balance between data utility and privacy. To the best of our knowledge, this is the first work where the parameter p in $\mathcal{L}_p$ norm is learnt inherently during the training process for privacy

preservation.

In the literature, researchers have shown the transferability of adversarial perturbations across different deep models (Demontis *et al.*, 2019; Papernot *et al.*, 2016a; Wu *et al.*, 2018b; Xie *et al.*, 2019). This ensures the generalizability of our proposed algorithm across unseen face-recognition models. The proposed PrIdentity algorithm does not require any target image for identity anonymization. Further, the proposed algorithm is extended for anonymizing multiple images via learning a single uniform perturbation, which could be a use case for suppressing an identity with multiple images of a given user on the social media platform. It is important to note that the aim of the proposed algorithm is to fool identity recognition networks and not humans. Multiple experiments are performed on the LFW (Huang *et al.*, 2008), CelebA (Liu *et al.*, 2015a), and CelebA-HQ (Karras *et al.*, 2018) datasets and a user study has been done to showcase the efficacy of the proposed algorithm.

3.2 Related Work

In the face-based privacy preserving literature, researchers have proposed various methods to suppress (i) attributes such as gender and age (Chhabra *et al.*, 2018; Mirjalili *et al.*, 2018; Shen *et al.*, 2019; Mirjalili *et al.*, 2020; Rana *et al.*, 2023) or (ii) facial identity for privacy preservation (Xiao *et al.*, 2020; Chen *et al.*, 2021a; Cai *et al.*, 2024; Xiao *et al.*, 2024; Li *et al.*, 2023a; Yang *et al.*, 2024a). The focus of this research is identity anonymization where existing work can be classified into seven broad categories: 1) k-anonymity, 2) morphing, 3) generative modeling, 4) reversible face anonymization 5) adversarial perturbations 6) Deepfakes 7) and non-generative modeling.

***k*-Anonymity:** Newton *et al.* (2005) have shown that ad-hoc methods such as obfuscating eyes are not able to thwart facial recognition systems. Therefore, the authors have proposed a k-anonymity-based algorithm (Samarati and Sweeney, 1998) termed as k-same that guarantees privacy preservation with the factor $\frac{1}{k}$. Gross *et al.* (2005) proposed a k-same select algorithm, which preserves the gender and expression in the de-identified images using a data utility function along with providing the guarantee of privacy preservation. The better quality of de-identified images is produced using Model-

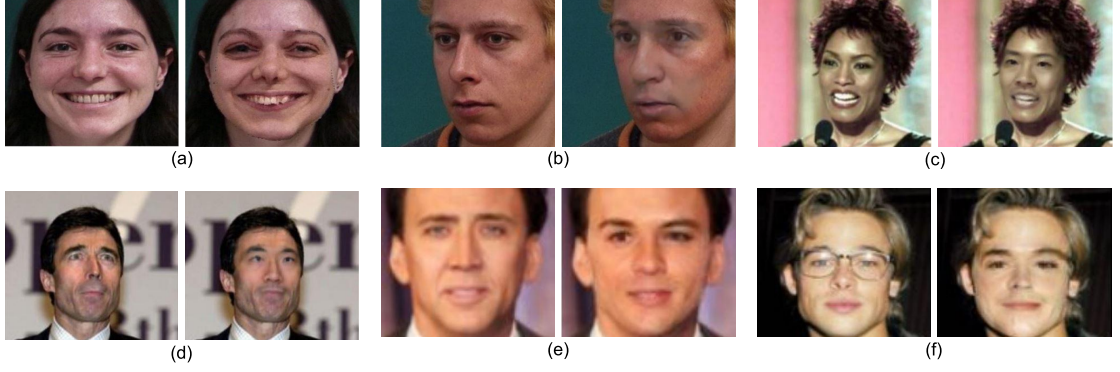


Figure 3.2: Sample identity-anonymized images are taken from the existing literature. In each pair, the image on the left represents the original, while the one on the right represents the anonymized version. “Images in (a) are from Gafni *et al.* (2019), images in (b) are generated through CIAGAN (Maximov *et al.*, 2020), images in (c) are generated through RiDDLE (Li *et al.*, 2023a), images in (d) are generated through FALCO (Barattin *et al.*, 2023), images in (e) are generated through FIT (Gu *et al.*, 2020), and images in (f) are generated through G²Face (Yang *et al.*, 2024a).

based k -Same algorithm, which is based on Active Appearance Models (Cootes *et al.*, 2001) and mixtures of identity and non-identity components in face images. Jourabloo *et al.* (2015) have proposed an attribute-preserving face de-identification technique that uses gradient descent to learn the weights for fusing k images.

Morphing: The techniques mentioned above, i.e., the k -same algorithm and their variants, replace original images with a set of de-identified images in an irreversible manner. Korshunov and Ebrahimi (2013) proposed a morphing-based method for visual privacy preservation. In this method, facial key points such as the nose and eyes are detected in both source and target images, and interpolation of both the keypoints is computed using a weighted sum of intensities, indicating the strength of privacy. In their other work (Korshunov and Ebrahimi, 2013), the image warping technique is used for privacy preservation. The authors have performed experiments on the Yale dataset. They also studied different privacy filters (Korshunov and Ebrahimi, 2014) such as blurring, pixelation, morphing, masking, and warping with varying strength degrees to investigate its effect on different face recognition algorithms. They observed that the morphing-based filter performs best for privacy preservation.

Generative Modeling: Various techniques have been proposed for face de-identification via generation face approaches (Li and Lin, 2019; Mirjalili *et al.*, 2020; Yang *et al.*,

Table 3.1: Comparison of existing algorithms for identity anonymization.

Paper	Method	Face generation type	Artifacts perceptible to humans	Attributes affected in face image	Multiple Image Anon.	User Study	Target Identity based Anon.
LIVE Gafni <i>et al.</i> (2019)	Adversarial Autoencoder coupled with trained face classifier	Visually similar faces	Visible	Nose, Eye, Facial texture	No	Yes	Yes
CIAGAN Maximov <i>et al.</i> (2020)	Conditional Generative Adversarial Net	Visually dissimilar and diverse set of images	Visible	Nose, Eyes, Makeup, Mouth region,	No	No	Yes
RIDDLE Li <i>et al.</i> (2023a)	StyleGAN inversion encoder coupled with Latent Encoder	Visually dissimilar and diverse set of images	Visible	Nose, Eyes, Mouth region	No	No	Yes
FALCO Barattin <i>et al.</i> (2023)	StyleGAN inversion coupled with ViT-based FaR	Visually dissimilar and diverse set of images	Visible	Hair, Nose, Eye, Mouth Region	No	No	Yes
FIT Gu <i>et al.</i> (2020)	Conditional GAN based Face Identity Transformer	Visually dissimilar and diverse set of images	Visible	Nose, Eyes, Mouth region, Face texture	No	Yes	Yes
G2Face Yang <i>et al.</i> (2024a)	StyleGAN coupled with 3D geometric priors	Visually dissimilar and diverse set of images	Visible	Nose, Eyes, Mouth region	No	No	Yes
Proposed + $\mathcal{L}_p$	Adversarial perturbation masks identity via p-norm optimization	Generate visually similar image	Not visible	No attributes are affected	Yes	Yes	No

2020). Meden *et al.* (2017) used a generative network to synthesize artificial surrogate faces for anonymity. The authors proposed a technique to preserve the non-identity-related information in the output face image. Wu *et al.* (2018d) proposed a Privacy-Protective-GAN framework for face de-identification. The proposed framework consists of two modules, the verifier with contrastive loss and the regulator with Structural Similarity Index (SSIM). The trade-off between privacy and image quality has also been shown in the results.

To improve the trade-off between privacy and data utility with improving the naturalness of the generated image through Style GAN, a clustering-based scheme is proposed by Le *et al.* (2020). Gafni *et al.* (2019) proposed an adversarial autoencoder network for real-time de-identification in the videos. The proposed network takes the source video as input and the generated de-identified output video has a similar-looking person. This network comes with a major restriction that it changes the skin-tone of the face in the output video and cannot preserve the original texture. Maximov *et al.* (2020) proposed a Conditional Identity Anonymization Generative Adversarial Network (CIA-GAN) for identity anonymization in both images and videos. In their proposed technique, trained GAN generates fake identities which are not present in the training set.

Chen *et al.* (2021c) used DCGANs to generate a similar-looking face image to capture all vital face attributes. Since the images are generated, the by-product is that the identity is masked. A similar work from Lin *et al.* (2021) focused on face de-

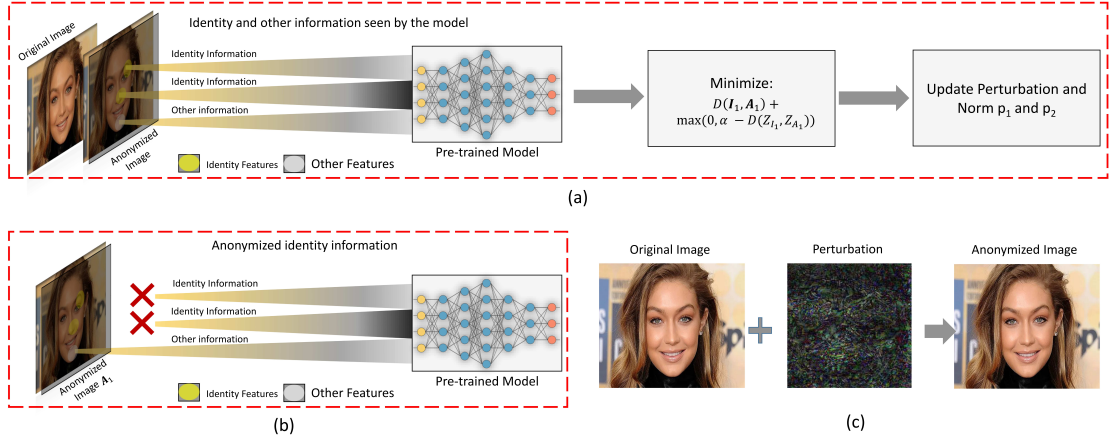


Figure 3.3: Illustrating the pipeline of the proposed algorithm. (a) shows the training process for learning a perturbation to generate an identity anonymized image. (b) shows that the pre-trained model is unable to extract the identity information from the anonymized image. (c) shows the addition of learned perturbation to obtain identity anonymized image.

identification through an improved version of UNet as the generator while using two discriminators. This architecture is termed FPGAN, which is trained using a weighted combination of pixel loss, content loss, and adversarial loss. The authors mathematically prove the convergence of the proposed architecture. Pan *et al.* (2021) took a leap and proposed an architecture that is capable of cyclic face-identification and re-identification. The authors use two encoders to learn style and content and a decoder network. This architecture is termed as Multi-factor Modifier (MfM), which achieves multi-factor facial de/re-identification.

Shamshad *et al.* (2023) proposed a two step method to search for adversarial latent codes and used StyleGAN to generate the anonymized image. The proposed method make changes in the makeup of the face to fool the models. Wen *et al.* (2023) proposed a IDEudemon method which uses a divide and conquer strategy for face de-identification. The proposed method obfuscate the 3D disentangled ID code using NeRF model and further used face parsing maps and GAN for identity anonymization. This method changes the geometry of identity related features eyes, nose, ears, mouth, and facial bones. Lopez *et al.* (2024) proposed a hardware level face anonymization, where the proposed method learns the optical encoder to generate the face heatmap and generates a new anonymized face image using a reference face image and face heatmap. The work by Laishram *et al.* (2024) used the StyleGAN and pretrained encoder to generate the de-identified caricature faces. The proposed method generates the caricature face for face de-identification. It uses the reference image for face de-identification. Re-

cently, Thakral *et al.* (2024) proposed a GAN based approach named 'ToonerGAN', which creates cartoon images for face de-identification. The proposed method transforms the image using a three step process i.e., Style Learning, Style Distillation, and De-identification. Although, these methods are able to achieve good quantitative results for face de-identifications but these methods suffers from preserving the original facial attributes such as eye color, shape of nose, skin tone, facial expression etc.

Reversible Face Anonymization: In the work from Wen *et al.* (2022), an IdentityMask framework is proposed for reversible face de-identification in videos. This framework uses deep motion flow for avoiding per-frame evaluation. Kuang *et al.* (2024a) proposed two-stage latent representation reorganization (LReOrg) framework for privacy preservation which uses bidirectional network optimized using keyword-based swap training strategy with a multi-task loss. This method changes the visual appearance of identity related features including nose, eyes. expressions etc. Li *et al.* (2023a) proposed RiDDLE for face de-identification. The RiDDLE use encryption decryption based scheme to recover the original images. Zhang *et al.* (2024) proposed another work FaceRSA, which is on similar lines of encryption decryption for face anonymization and recovery. Yang *et al.* (2024a) proposed a reversible face anonymization method termed as G²Face. The proposed methods extracts geometric information from 3D face and use generative priors in a multi-scale manner for reversible face anonymization. This method changes identity related features such as nose, mouth shape, shape of eyebrows etc. These works are primarily inspired from GAN based methods identity anonymization. Figure 3.2 shows some sample identity anonymized images from existing literature. Table 3.1 shows the comparison of existing algorithms with the proposed algorithm for identity anonymization.

Adversarial Perturbations: In the literature, few methods used adversarial perturbations for identity anonymization. Li and Lin (2019) proposed a four stage framework called AnonymousNet. The framework consists of facial attribute estimation, privacy-metric-oriented face obfuscation, directed natural image synthesis, and adversarial perturbation as the four stages. A targeted identity-protection iterative method (TIP-IM) via generating identity perturbation masks is proposed by Yang *et al.* (2020). For anonymizing the identities, TIP-IM generates the perturbation masks while ensuring the transferability of the generated identity anonymized image. The training and generation of masks is done in a white-box setting while the testing and evaluation is

done in a black-box setting. Zhong and Deng (2022) used adversarial perturbations to generate person specific mask. However, the authors used 10 images to generate the privacy mask, which may not be available in real world scenarios. Liu *et al.* (2024) addresses the optimization based problem for class specific face de-identification. The authors discovered the optimization problem in large and small batch optimization and proposed the Gradient Accumulation (GA) for gradient stability. There is no visual output of the de-identified images are shown in the paper. Yang *et al.* (2024c) proposed a face de-identification method, which is primarily applicable in cloud platforms only for storing images in secured manner. The proposed method transforms the images such that it is unrecognizable to humans and only recognizable to face recognition models.

Deepfakes: In the literature, researchers also leverage the methods used to generate Deepfakes for identity anonymization. In the work from Zhu *et al.* (2020), a Deepfake based technique is used in medical domain to examine Parkinson’s disease through movement in facial key points. The authors used faceswap for the de-identification of faces while keeping facial key points invariant. Cai *et al.* (2024) proposed Disguise method for face de-identification which is based on a face swapping method. This method changes the visual appearance of identity related attributes including nose, eyes, expressions etc.

Non Generative Modeling: Kuang *et al.* (2024b) proposed a new method for face identity anonymization, which distracts the intrinsic identity information in feature space and extrinsic identity information such as visual cues in visual space. The proposed method changes the appearance of identity related attributes such as eye color, nose shape etc. To preserve privacy in face datasets, Singh *et al.* (2022) proposed a two-step sanitizing framework for the privacy-preservation of the entire dataset. In the first step, a global decoupling process decomposes raw data into sensitive and non-sensitive latent representations. The next step consists of local sampling to synthetically generate sensitive information with differential privacy and merge it with non-sensitive latent features while preserving privacy. Jiang *et al.* (2023) proposed the DartBlur method for privacy preservation, which generates blurred faces using DNN. Yuan *et al.* (2024) proposed a client server based framework PRO-Face C, which performs face recognition in a privacy preserving manner. The proposed method obfuscate the image on client side and use feature compensation mechanism to optimize the equilibrium between privacy preservation and utility maximization on server side.

Based on the existing literature, we observed that the majority of the existing algorithms end up in modifying the identity related facial attributes such as shape of nose, eye color, facial color etc. These changes are easily noticeable to the users, which defeats the purpose of identity anonymization for social media applications, where the visual appearance of the original and anonymed images should be similar. Also, the existing methods are not able to maintain a good balance between privacy and data utility. In addition, there are limited existing algorithm with several limitations that can anonymize multiple input images using a single uniform perturbation.

3.3 Privacy Preserving Identity Anonymization

Existing approaches are primarily dependent on the target face images that are used to fuse with the input face image for identity anonymization. This results in visually distorted images and loss of data utility. Other approaches modify the skin tone and change the nose structure and eye color to suppress identity information, which hinders the applicability for different applications. Therefore, in this research, we have proposed a novel privacy-preserving identity anonymization or PrIdentity algorithm that ensures privacy, data utility, and generalizability across different unseen face recognition models.

The proposed PrIdentity algorithm perturbs the input face image to suppress the identity information and preserve the visual appearance of the perturbed image. In order to maintain a trade-off between data utility and privacy, we have used the $\mathcal{L}_p$ norm-based regularization technique for learning adversarial perturbation, where p in $\mathcal{L}_p$ norm is a learnable parameter. We have further extended the proposed algorithm to anonymize multiple images simultaneously via learning a single uniform perturbation. Figure 3.3 illustrates the pipeline of the proposed algorithm which can be interpreted as: "Learn a perturbation $\mathbf{P}$ for an input image $\mathbf{I}$ such that on adding perturbation $\mathbf{P}$ to an input image $\mathbf{I}$, the output image $\mathbf{A}$ is identity anonymized while maintaining the balance between data utility and privacy". The detailed steps for the proposed algorithm are discussed below.

Let $\mathbf{S} = \{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_m\}$ be a subject with m number of images. Let $\mathbf{I}_k$ be any image of subject $\mathbf{S}$ with pixel values in the range of 0 to 1, i.e., $\mathbf{I}_k \in [0, 1]$. Let $\mathbf{P}$ be the

adversarial perturbation to be learned. As mentioned in the problem statement, the task is to learn a perturbation $\mathbf{P}$ such that on adding $\mathbf{P}$ to an input image $\mathbf{I}_k$, the output image $\mathbf{A}_k$ is identity anonymized. In other words, on adding $\mathbf{P}$ to $\mathbf{I}_k$, the class/identity of the output anonymized image $\mathbf{A}_k$ should be different from the class/identity of the input image $\mathbf{I}_k$. Mathematically, it is written as:

$$\mathbf{A}_k = \mathbf{I}_k + \mathbf{P} \quad (3.1)$$

$$\text{such that } C(\mathbf{A}_k) \neq C(\mathbf{I}_k)$$

where $C(\cdot)$ is the function that outputs the identity of an image. Let $\phi(\mathbf{W}, \mathbf{b})$ be any pre-trained face recognition model that takes an image as an input and outputs its identity representation vector. Thus, the identity representation vector corresponding to an input image $\mathbf{I}_k$ and anonymized image $\mathbf{A}_k$ is obtained as:

$$\mathbf{Z}_{\mathbf{I}_k} = \phi(\mathbf{W}\mathbf{I}_k + \mathbf{b}) \text{ and } \mathbf{Z}_{\mathbf{A}_k} = \phi(\mathbf{W}\mathbf{A}_k + \mathbf{b}) \quad (3.2)$$

where, $\mathbf{Z}_{\mathbf{I}_k}$ and $\mathbf{Z}_{\mathbf{A}_k}$ are the identity representation vector. $\mathbf{W}$ is the weight matrix and $\mathbf{b}$ is the bias of the model ϕ .

In order to anonymize the identity in an input image $\mathbf{I}_k$, the distance between $\mathbf{Z}_{\mathbf{A}_k}$ and $\mathbf{Z}_{\mathbf{I}_k}$ is maximized. Mathematically, it is written as:

$$\text{maximize } D(\mathbf{Z}_{\mathbf{I}_k}, \mathbf{Z}_{\mathbf{A}_k}) \quad (3.3)$$

where, $D(\cdot)$ is the distance metric. The above equation corresponds to the privacy measure of the input image $\mathbf{I}_k$ and satisfies the constraint of Equation 3.1.

In many real-world applications, it is important that the visual appearance of the image should not be distorted. Therefore, to preserve the visual appearance of an anonymized image, the distance between the anonymized image $\mathbf{A}_k$ and input image $\mathbf{I}_k$ is minimized. Mathematically, it is written as:

$$\text{minimize } D(\mathbf{I}_k, \mathbf{A}_k) \quad (3.4)$$

where, Equation 3.4 corresponds to the data utility measure of the input image $\mathbf{I}_k$. Using

Table 3.2: Details of the experiments performed to evaluate the performance of the proposed algorithm.

Experiment	Dataset	Evaluation Type	Gallery Size	Description
Single Image Anonymization	CelebA	Face Identification	1, 2, 3, 4	This experiment is performed to showcase the performance of the proposed algorithm in face identification setting. There is no literature available for the comparison.
	LFW	Face Verification	NA	This experiment is performed to showcase the performance in face verification setting. Evaluation on the LFW dataset in the verification setting is the standard protocol followed in the literature for the comparison.
Multiple Image Anonymization	CelebA	Face Identification	2, 4	This experiment is performed to showcase the performance of the proposed algorithm in face identification setting. There is no literature available for the comparison.
Bounding Box Prediction	CelebA-HQ	Face Detection	NA	This experiment is performed to showcase the performance of the proposed algorithm for face detection. Evaluation on the CelebA-HQ dataset for face detection is the standard protocol followed in the literature for comparison.

Equation 3.3 and 3.4, the objective function is written as:

$$\text{minimize } D(\mathbf{I}_k, \mathbf{A}_k) - D(\mathbf{Z}_{\mathbf{I}_k}, \mathbf{Z}_{\mathbf{A}_k}) \quad (3.5)$$

The perturbation $\mathbf{P}$ is learned by minimizing the above objective function. In order to control the separation among the identity representation $\mathbf{Z}_{\mathbf{A}_k}$ and $\mathbf{Z}_{\mathbf{I}_k}$, a margin parameter α is introduced. The margin parameter α ensures that the distance between $\mathbf{Z}_{\mathbf{A}_k}$ and $\mathbf{Z}_{\mathbf{I}_k}$ should be at least α . Thus, the above objective function is updated as:

$$\text{minimize } D(\mathbf{I}_k, \mathbf{A}_k) + \max(0, \alpha - D(\mathbf{Z}_{\mathbf{I}_k}, \mathbf{Z}_{\mathbf{A}_k})) \quad (3.6)$$

As mentioned, the proposed PrIdentity algorithm should maintain the trade-off between privacy and data utility and ensure generalizability across different unseen face recognition models. The details for maintaining trade-offs and ensuring generalizability are described in the following subsections.

3.3.1 Data Utility vs Privacy

In many real-world applications, maintaining a good balance between data utility and privacy is of paramount factor. For this purpose, we have used the p - norm function as

a distance metric, where p is a parameter to be learned. Thus, the objective function is updated as:

$$\begin{aligned} & \text{minimize} \left(\left(\sum_{x,y} \|\mathbf{I}_k(x, y) - \mathbf{A}_k(x, y)\|^{p_1} \right)^{\frac{1}{p_1}} + \right. \\ & \left. \max \left(0, \alpha - \left(\sum_{i=1}^d \|\mathbf{Z}_{\mathbf{I}_k, i} - \mathbf{Z}_{\mathbf{A}_k, i}\|^{p_2} \right)^{\frac{1}{p_2}} \right) \right) \end{aligned} \quad (3.7)$$

where, d is the dimension of the identity representation vectors $\mathbf{Z}_{\mathbf{I}_k}$ and $\mathbf{Z}_{\mathbf{A}_k}$. p_1 and p_2 are the parameters corresponding to the privacy and data utility measure, respectively. In the literature, $\mathcal{L}_1$ norm where $p = 1$ and $\mathcal{L}_2$ norm where $p = 2$ are widely used in machine learning and deep learning applications. However, it is challenging to decide the suitable value of p -norm for a given application. To overcome this challenge, in this research, we have treated $p \in [1, 2]$ as a parameter and learned the optimal p to balance the privacy and data utility. To learn the value of p , we have used the SGD optimizer where we randomly initialize its value in the range $[1, 2]$ and use the clip function to bind its value.

3.3.2 Generalizability

To ensure generalizability, it is important that the learned perturbation must be transferable to other deep models. In the literature, researchers have shown that the iterative learning of a perturbation generally overfits the surrogate model and hence performs poorly when evaluated on the target model. In general, the $\mathcal{L}_2$ norm is sensitive to outliers and is prone to overfitting. On the other hand, the $\mathcal{L}_1$ norm underfits the target model while generating adversarial perturbations. In our proposed algorithm, we address this problem by learning the optimal value of p in the range $[1, 2]$, which results in robust perturbation that is transferable to unseen target face-recognition models. There can be two cases for identity anonymization:

Case I - Single Image Anonymization: This case corresponds to the scenario, where a user anonymizes a single image only. In this case, the perturbation will be different for each image for a given user and therefore, it provides more control to the user over the privacy of each image. For single image anonymization, we use $k = 1$ and

Equation 3.6 is updated as:

$$\text{minimize } D(\mathbf{I}_1, \mathbf{A}_1) + \max(0, \alpha - D(\mathbf{Z}_{\mathbf{I}_1}, \mathbf{Z}_{\mathbf{A}_1})) \quad (3.8)$$

where, first term is minimizing the distance between original and anonymized images while the second term is maximizing the distance between the face representations of the original and anonymized images.

Case II - Multiple Image Anonymization: This case corresponds to the scenario, where a user anonymizes multiple images of the same user. In this scenario, a single uniform perturbation $\mathbf{P}$ is learned for all the input images. Let n be the number of images to be anonymized. Thus, the objective function is written as:

$$\text{minimize } \sum_{k=1}^n \{D(\mathbf{I}_k, \mathbf{A}_k) + \max(0, \alpha - D(\mathbf{Z}_{\mathbf{I}_k}, \mathbf{Z}_{\mathbf{A}_k}))\} \quad (3.9)$$

Here, the above Equation iterated over the multiple images of the same user. In this case, the user has comparatively less control over the privacy as the perturbation will be the same for n images of a given user. However, this use case has relatively computationally efficient compared to single image anonymization.

3.4 Experimental Setup

In order to evaluate the performance of the proposed PrIdentity algorithm, two experiments are performed: 1) Single Image Anonymization and 2) Multiple Image Anonymization. The first experiment represents the scenario where only a single image for each identity is anonymized. In this experiment, images are anonymized by learning and adding a perturbation for each image. The second experiment represents the scenario where multiple images for each identity are anonymized simultaneously. In this experiment, a single uniform perturbation is learned to anonymize multiple images of the same identity. The comparison is performed with existing state-of-the-art algorithms. All the experiments are performed for $p = 1$, $p = 2$, and $1 < p < 2$ i.e $\mathcal{L}_1$, $\mathcal{L}_2$, and

Table 3.3: Identification accuracy (%) for single image anonymization experiment on the original and identity anonymized images obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm on the CelebA dataset. To showcase the genrealizability of the proposed algorithm, the results are reported on VGGFace, ArcFace, and LightCNN-29 models respectively. A low accuracy indicates a high performance (better privacy).

Gallery Size 1												
	Original			Proposed + $\mathcal{L}_1$			Proposed + $\mathcal{L}_2$			Proposed + $\mathcal{L}_p$		
	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10
VGGFace	45.42	61.71	68.04	0.20	0.70	1.40	0.20	0.80	1.70	0.30	0.80	1.30
ArcFace	64.46	76.66	80.94	5.40	14.03	19.71	2.93	8.52	12.99	2.55	7.50	11.44
LightCNN	83.91	90.53	92.43	18.04	32.52	40.31	10.60	22.69	29.09	9.57	19.83	26.17
Gallery Size 2												
	Original			Proposed + $\mathcal{L}_1$			Proposed + $\mathcal{L}_2$			Proposed + $\mathcal{L}_p$		
	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10
VGGFace	57.30	73.27	79.05	0.10	0.67	1.34	0.15	0.72	1.18	0.25	0.72	1.030
ArcFace	76.17	87.2	90.27	7.84	18.13	24.86	3.90	10.76	16.08	3.19	9.18	13.94
LightCNN	91.39	95.63	96.48	24.00	44.12	51.37	13.67	27.68	36.30	12.61	26.32	33.95
Gallery Size 3												
	Original			Proposed + $\mathcal{L}_1$			Proposed + $\mathcal{L}_2$			Proposed + $\mathcal{L}_p$		
	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10
VGGFace	63.16	78.25	83.55	0.14	0.42	0.94	0.14	0.52	0.98	0.14	0.38	0.98
ArcFace	81.74	90.55	93.09	90.00	20.98	28.65	4.46	11.59	17.07	3.40	10.35	15.52
LightCNN	93.26	96.32	97.08	27.95	47.78	56.83	17.01	32.87	41.36	14.94	30.36	38.71
Gallery Size 4												
	Original			Proposed + $\mathcal{L}_1$			Proposed + $\mathcal{L}_2$			Proposed + $\mathcal{L}_p$		
	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10
VGGFace	87.33	81.44	86.23	0.16	0.61	1.31	0.16	0.51	0.96	0.21	0.51	0.99
ArcFace	84.27	92.13	94.31	10.03	22.64	30.58	5.25	13.65	19.67	3.92	11.09	16.99
LightCNN	94.14	96.88	97.51	30.84	51.53	60.83	19.03	36.41	45.59	16.94	32.76	42.24

$\mathcal{L}_p$ norm, respectively. Here, $\mathcal{L}_p$ norm represents the case where the parameters p_1 and p_2 corresponding to privacy and data utility (visual appearance) are learned. Table 3.2 shows the details of the experiments. To quantify the data utility of the proposed algorithm, we have computed the bounding box distance on the CelebA-HQ dataset and compared the results with the existing algorithms.

To showcase the effectiveness of the proposed PrIdentity algorithm from the user’s perspective, we have also conducted a user study on the original and their corresponding perturbed images. More details and results of the user study are discussed in section 3.5.4. Next, we discuss the data sets and protocol in detail. All the datasets used for evaluation in this research are publicly available to the research community.

LFW dataset (Huang *et al.*, 2008): This dataset consists of 13,233 images of 5,749 subjects. To evaluate the performance of the proposed PrIdentity algorithm for face verification, standard view 2 of this dataset consisting of 6000 image pairs is used. One image from each pair is considered for identity anonymization.

Table 3.4: Data utility (SSIM) scores for single image anonymization experiment on the original and corresponding identity anonymized images obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm on the CelebA dataset. A high score indicates high performance.

Gallery Size	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_p$
1	0.9004	0.8629	0.84187
2	0.9001	0.8628	0.8422
3	0.9000	0.8628	0.8423
4	0.8999	0.8625	0.8414

CelebA dataset (Liu *et al.*, 2015a): This dataset contains 202,599 face images of more than 10k celebrities. The performance of the proposed algorithm is evaluated for the face identification task. We have performed experiments on the test set, which consists of around 1k identities. The images of each subject are divided into the gallery and probe sets. Only gallery-set images are considered for identity anonymization. A single-image anonymization experiment is performed with a gallery set having 1, 2, 3, and 4 images corresponding to each identity, while a multiple-image anonymization experiment is performed with a gallery set having 2 and 4 images.

CelebA-HQ dataset (Karras *et al.*, 2018): This dataset comprises of 30,000 high-quality images. The images in this dataset are the high-quality version of the CelebA dataset. We used these images to generate the identity anonymized images using the proposed algorithm and further compared the face detection performance on original and identity anonymized images.

Face images from all three datasets are detected using Multi-Task Cascaded Convolutional Neural Networks (MTCNN) (Zhang *et al.*, 2016). To measure the anonymized images’ data utility, we have computed the Structural similarity index measure (SSIM) (Wang *et al.*, 2004) score for each anonymized image corresponding to the given original input image. For measuring the data utility U of the complete dataset with anonymized images, we have taken the average of the SSIM score. The data utility score 1 represents that the image’s visual appearance is completely preserved, while the data utility score 0 represents the image is completely distorted. A given dataset’s privacy score

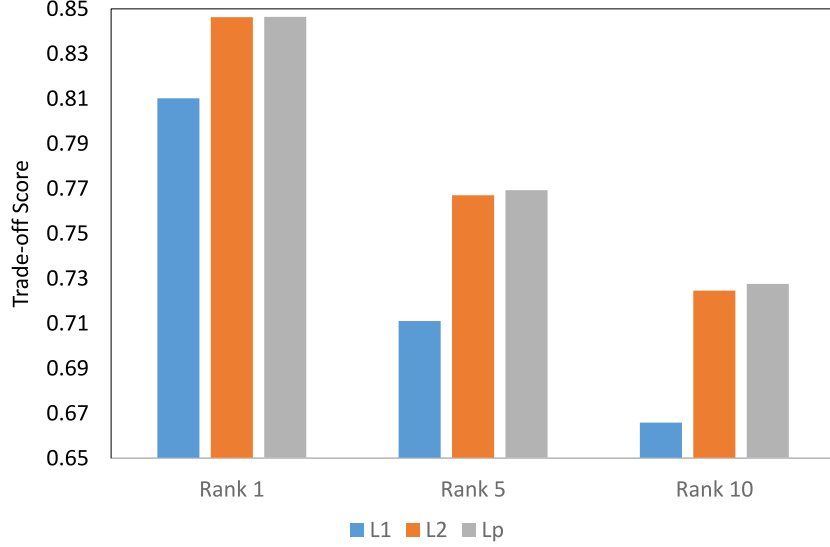


Figure 3.4: Comparison of trade-off score (T) for single image anonymization experiment corresponding to gallery size 3 on the CelebA dataset.

Pr is measured by subtracting and dividing the obtained accuracy on the given dataset by 100. This means if the obtained accuracy is 100%, then the privacy score would be $\frac{100-100}{100}$, i.e., 0. On the other hand, if the obtained accuracy is 0%, then the privacy score would be $\frac{100-0}{100}$, i.e., 1. Ideally, both data utility U and privacy score Pr should be 1. To quantify the trade-off between data utility and privacy, we have taken the mean of data utility and privacy score. Mathematically, it is written as:

$$T = \frac{U + Pr}{2} \quad (3.10)$$

where, U and Pr represents the data-utility and privacy score, while T represents the trade-off score.

All images are anonymized corresponding to the ResNet50 (He *et al.*, 2016) model pre-trained on the VGGFace2 (Cao *et al.*, 2018) dataset for both the experiments. To evaluate the proposed algorithm's generalizability across different unseen face recognition models, we have reported the recognition performance using VGGFace (Parkhi *et al.*, 2015), ArcFace (Deng *et al.*, 2019), and LightCNN-29 (Wu *et al.*, 2018c). The implementation details of the proposed algorithm are discussed in the following subsection.

Table 3.5: Identification accuracy (%) for multiple image anonymization experiment on the original and identity anonymized images obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm on the CelebA dataset. To showcase the generalizability of the proposed algorithm, the results are reported on VGGFace, ArcFace, and LightCNN-29 models respectively. A low accuracy indicates a high performance (better privacy).

Gallery Size 2												
	Original			Proposed + $\mathcal{L}_1$			Proposed + $\mathcal{L}_2$			Proposed + $\mathcal{L}_p$		
	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10
VGGFace	57.30	73.27	79.05	0.10	0.61	1.44	0.20	0.77	1.18	0.25	0.56	1.13
ArcFace	76.17	87.2	90.27	20.62	38.21	46.83	7.31	17.67	24.17	5.38	13.7	19.91
LightCNN	91.39	95.63	96.48	47.93	67.67	74.93	24.7	42.44	50.81	21.2	38.15	46.63
Gallery Size 4												
	Original			Proposed + $\mathcal{L}_1$			Proposed + $\mathcal{L}_2$			Proposed + $\mathcal{L}_p$		
	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10	Rank 1	Rank 5	Rank 10
VGGFace	67.33	81.44	86.23	0.10	0.56	1.07	0.10	0.53	1.05	0.18	0.64	1.15
ArcFace	84.27	92.13	94.31	42.22	63.08	71.28	11.49	24.56	32.9	8.35	19.58	27.29
LightCNN	94.14	96.88	97.51	73.45	88.51	91.21	37.80	61.84	68.14	32.48	53.06	61.92

3.4.1 Implementation Details

All the experiments are performed on a DGX station with Intel Xeon CPU, 256 GB RAM, and four 32 GB Nvidia V100 GPU cards. The implementation details for learning perturbation and evaluating performance on different face recognition models are given below.

Perturbation learning: To learn a perturbation for both experiments, the proposed algorithm is implemented in Tensorflow 2.3.1. The perturbation is learned corresponding to the ResNet50 model pre-trained on the VGGFace2 dataset. The learning rate is set to 0.1 during training, and the perturbation is trained for 120 iterations. The margin α is set to 2 during the experiments.

For evaluating performance using the VGGFace model (Parkhi *et al.*, 2015), the original and anonymized images are resized to 224×224 . While evaluating performance using ArcFace (Deng *et al.*, 2019) and LightCNN-29 (Wu *et al.*, 2018c) models, features are extracted after resizing the images to 128×128 resolution grayscale images. The distance between features of images is computed using cosine similarity.

3.4.2 Fair Comparison with Existing Methods

In the recent literature of face identity anonymization, it has been observed that the researchers are making unfair comparison with the existing methods to showcase state of

the results. For example, ToonerGAN (Thakral *et al.*, 2024), which generates the caricature/cartoon images achieved state-of-the-art results on the LFW dataset by achieving 0.009 TPR at FPR of 0.001 and showed comparison with existing methods such as CIAGAN (Maximov *et al.*, 2020), which generates photo realistic anonymized images having some modifications in identity related attributes such as eye and nose. In order to make a fair comparison of privacy quantitatively, it is important that the use case or data utility of the existing methods should be similar. In this research, we have anonymized the face images for social media based applications, where the automated face recognition algorithms should not be able to recognize the anonymized image of an user uploaded on the social and the user perceives an anonymized image on social media as real image.

In order to make a fair comparison, we have compared the proposed algorithm with the state-of-the-art (sota) GAN based methods such as CIAGAN (Maximov *et al.*, 2020), RIDDLE (Li *et al.*, 2023a), LIVE (Gafni *et al.*, 2019), FIT (Gu *et al.*, 2020), FALCO (Barattin *et al.*, 2023), and G2Face (Yang *et al.*, 2024a) for identity anonymization. Although, these methods modifies the identity related attributes such as shape of nose, eye color, facial expression etc and are not perfectly suitable of the social media use case but these methods are the closest ones to compare. We have also compared the bounding box distance of the proposed algorithm against these methods to showcase the data utility of the proposed algorithm.

3.5 Performance Evaluation

The quantitative results of the proposed PrIdentity algorithm are evaluated for single and multiple-image anonymization. The qualitative and quantitative comparison is performed with the existing algorithms. For qualitative evaluation, the performance obtained through user study is also reported for a combination of both single and multiple image anonymization. The results for all three experiments are discussed in the following subsections.

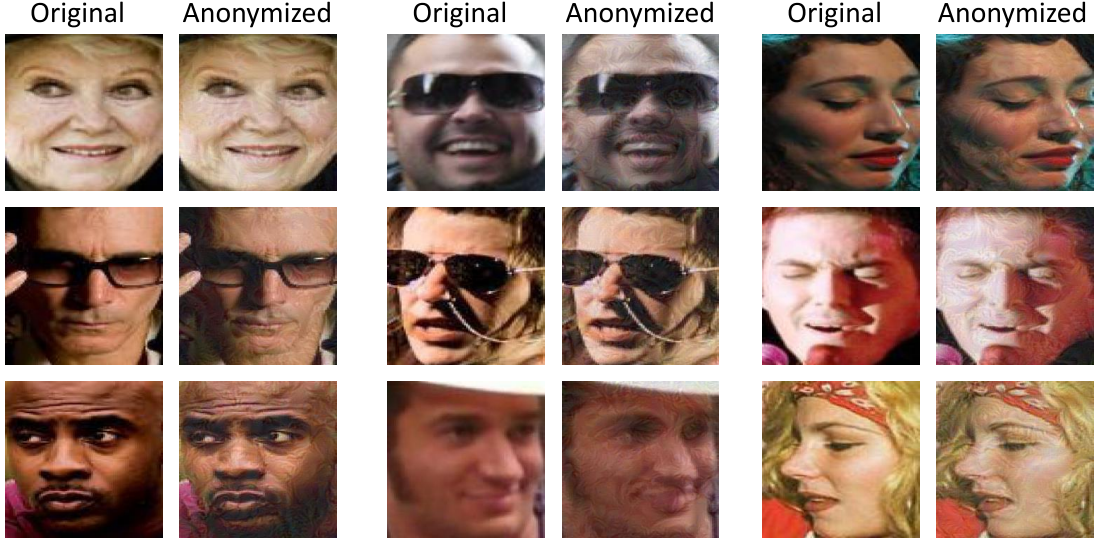


Figure 3.5: Sample images before and after anonymization obtained using the proposed algorithm $\mathcal{L}_p$ norm.

Table 3.6: Verification performance of the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm at FPR 0.001 on the LFW dataset.

	Original	$\mathcal{L}_1$	$\mathcal{L}_2$	$\mathcal{L}_p$
ArcFace	0.9940	0.0077	0.0047	0.002
LightCNN-29	0.9930	0.0127	0.0147	0.013

3.5.1 Single Image Anonymization

In this experiment, a perturbation is learned corresponding to each image for the identity anonymization task. The results for both CelebA and LFW datasets are discussed below.

Results on the CelebA dataset: Table 3.3 shows the identification accuracy on the original and identity anonymized images obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm respectively. The results in Table 3.3 correspond to the gallery set having 1, 2, 3, and 4 images for each identity. The results are reported at ranks 1, 5, and 10, respectively, using the VGGFace, ArcFace, and LightCNN-29 face recognition models. Rank-k denotes the model accuracies when correct match was within the model’s top-k predictions. Following are the observations from the results reported in Table 3.3.

1. On evaluating the proposed algorithm’s performance using VGGFace, ArcFace, and LightCNN-29 model (BlackBox evaluation), we observe that the proposed al-

gorithm is able to successfully anonymize the images, and results in low accuracy of the model. The identification performance is better on VGGFace model compared to ArcFace and LightCNN-29 models because ArcFace and LightCNN-29 models use grayscale images as an input and perturbation is learned use in color images. The overall performance of the algorithm shows that the proposed algorithm is able to preserve the privacy of an individual in real-world scenarios.

2. On comparing the performance of $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm, it is observed that the proposed algorithm with $\mathcal{L}_p$ norm is performing better at ranks 1, 5, and 10, especially for ArcFace and LightCNN-29 face recognition models. It shows that the learnable parameter p in $\mathcal{L}_p$ norm helps to achieve better privacy on the anonymized images. Also, it shows that the proposed algorithm with p norm is generalizable at higher ranks.
3. $\mathcal{L}_1$ norm introduces sparsity and $\mathcal{L}_2$ norm is sensitive to outliers and imposes convexity. Using either $\mathcal{L}_1$ norm or $\mathcal{L}_2$ norm results in low performance due to overfitting towards privacy or data utility. On the other hand, p in $\mathcal{L}_p$ norm is a learnable parameter in the range of 1 to 2. Therefore, it introduces an adequate amount of sparsity and convexity and is not sensitive to outliers, leading to better performance. More results on the comparison of $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm are discussed later.
4. In many real-world applications, multiple images of the same identity are stored in the gallery set. Therefore, the anonymization algorithm must be applicable with the multiple images per subject in the gallery set. On comparing the performance with the increasing number of images from 1 to 4 in the gallery set, we observe an increase in the accuracy of the ArcFace face recognition model from 64.46% to 84.27% on the original images. However, the performance on the identity anonymized images obtained using the proposed algorithm with $\mathcal{L}_p$ norm marginally increased by 1.37%. Similar patterns can also be observed for other models where the increase in the rank 1 accuracy on the identity anonymized images is relatively smaller to the increase in the accuracy on the original images. This shows the applicability of the proposed anonymization algorithm in real-world scenarios where multiple images are stored in the gallery set.
5. We have also observed that the LightCNN-29 model performs best on the original and identity anonymized images at ranks 1, 5, and 10. Overall, the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm is least effective on the LightCNN-29 model.

Effectiveness of $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm: In order to measure the effectiveness of individual norms for the proposed algorithm, it is important to evaluate the privacy score, data-utility score, and the trade-off between them. As mentioned above, lower accuracy represents better privacy. It implies that the $\mathcal{L}_p$ norm achieves better privacy compared to $\mathcal{L}_1$ and $\mathcal{L}_2$ norms. Table 3.4 shows the data utility scores obtained by computing SSIM scores. A higher value indicates better data utility. It is observed that the $\mathcal{L}_1$ norm has better data utility, followed by the $\mathcal{L}_2$ norm and $\mathcal{L}_p$ norm. However, the effectiveness of the algorithm should be measured by the trade-off between privacy and data

Table 3.7: Comparing the True Positive Rate of the proposed algorithm with state-of-the-art identity anonymization algorithms on the LFW dataset at 0.001 False Positive Rate.

	FaceNet	ArcFace
Original	0.965	0.987
LIVE (Gafni <i>et al.</i> (2019))	0.035	0.025
CIAGAN (Maximov <i>et al.</i> (2020))	0.019	0.021
FIT (Gu <i>et al.</i> (2020))	0.033	0.027
RIDDLE (Li <i>et al.</i> (2023a))	0.032	0.011
FALCO (Barattin <i>et al.</i> (2023))	0.021	0.015
G2Face (Yang <i>et al.</i> (2024a))	0.009	0.007
Proposed + $\mathcal{L}_p$	0.019	0.002



Figure 3.6: Comparison of the anonymized images generated using existing algorithms Gafni *et al.* (2019), Maximov *et al.* (2020) and the proposed algorithm. The images generated using existing algorithms are heavily distorted while the visual appearance of the images generated using the proposed algorithm is preserved.

utility. Figure 3.4 shows the comparison of the trade-off score of $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norms at ranks 1, 5, and 10 respectively, corresponding to the LightCNN-29 model for gallery set having 3 images for each identity. It is observed that the $\mathcal{L}_1$ norm is performing worst at ranks 1, 5, and 10. On the other hand, the $\mathcal{L}_p$ norm performs best at rank 5 and 10, while on rank 1, it shows comparable performance with the $\mathcal{L}_2$ norm. These quantitative results indicate that the $\mathcal{L}_p$ norm effectively maintains a good balance between privacy and data utility. This trade-off can also be qualitatively visualized in Figure 3.5.

Results on the LFW dataset and comparison with existing algorithms: To evaluate

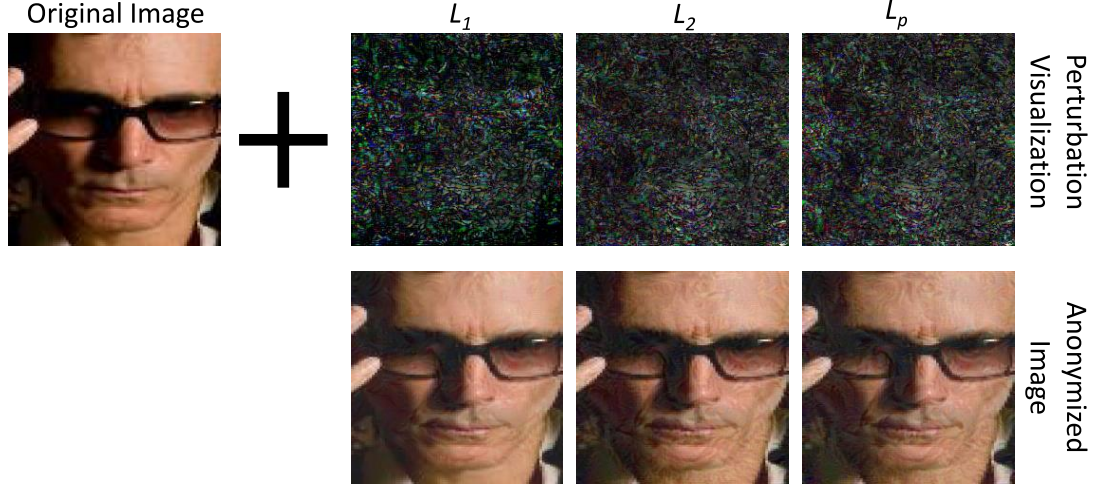


Figure 3.7: Sample images before and after anonymization obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm on the CelebA dataset.

the performance on the LFW dataset, one image from each pair is anonymized. The verification performance is evaluated on the anonymized images using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norms as shown in Table 3.6. We have observed that the proposed algorithm with $\mathcal{L}_p$ norm performs well and reduces the True Positive Rate to 0.013, corresponding to LightCNN-29 model. The better results on both face identification and verification tasks show the suitability of the proposed algorithm for different applications.

The results are also compared with state-of-the-art algorithms, as shown in Table 3.7. FaceNet model pre-trained on the CASIA-Webface database and ArcFace model trained on MS1MV3 dataset are used for the comparison. Our proposed algorithm with $\mathcal{L}_p$ norm achieves the second best result with 0.019 True Positive Rate when evaluated using the FaceNet model pre-trained on CASIA-Webface dataset (Yi *et al.*, 2014). It achieved state-of-the-art (sota) results with 0.002 True Positive Rate when evaluated with ArcFace model. We would like to mention that the proposed algorithm does not require the target image for the anonymization while maintaining the visual appearance, which is not the case with other compared algorithms. Figure 3.6 shows some anonymized sample images generated using existing algorithms and the proposed algorithms. It is clear from Figure 3.6 that the visual distortions are present in the images generated using CIAGAN (Maximov *et al.*, 2020) and Gafni *et al.* (2019). However, the visual appearance is preserved in the images generated using the proposed algorithm. CIAGAN uses an identity guidance discriminator, which requires target identity infor-

mation as input to generate the anonymized image. As a result, the image generated using CIAGAN has some characteristics of target identity and thus visually distorts the image. The method proposed by Gafni *et al.* (2019) and Yang *et al.* (2024a) requires the target image or dummy embedding to generate the anonymized image. However, our proposed algorithm does not require any target image or dummy embedding and generates visually natural images.

3.5.2 Multiple Image Anonymization

This experiment is performed to showcase that the proposed algorithm can also be extended for scenarios where the user wants to suppress multiple images simultaneously. In this experiment, multiple images of the same identity are anonymized via learning a single uniform perturbation. The experiments are performed by anonymizing 2 and 4 images of the same identity. Table 3.5 shows the identification accuracy on the original and identity anonymized images obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norms. Similar to the results of single image anonymization, the proposed algorithm with $\mathcal{L}_p$ norm is performing better when 2 and 4 images for each identity are anonymized in the gallery set via single uniform perturbation. The identification accuracy obtained for gallery set having 2 and 4 images corresponding to the VGGFace model at rank 10 is 79.05% and 86.23%, respectively, while after anonymization, the performance of the proposed algorithm with $\mathcal{L}_p$ norm is 1.13% and 1.15%. This shows the efficacy of the proposed algorithm.

It is important to note that the performance obtained using ArcFace and LightCNN-29 models are relatively lower compared to VGGFace model because the former models use grayscale images as an input while the perturbation is learned corresponding to color images. On comparing these results with a single image anonymization experiment for a gallery set having 2 and 4 images, it is observed that anonymizing a single image gives better results. However, anonymizing multiple images simultaneously is computationally efficient compared to single image anonymization. Overall, it's a tradeoff between privacy and computation efficiency when comparing the performance of single and multiple image anonymization results. Figure 3.7 shows some images obtained using the proposed algorithm with $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_p$ norm. It is clear from Figure 3.7 that the visual appearance of the images are preserved after anonymization.

Comparison with OPOM (Zhong and Deng, 2022): The method proposed by OPOM (Zhong and Deng, 2022) generates a person specific perturbation for identity anonymization. However, it has several limitations. OPOM uses 10 images of a person for identity anonymization, which is large and may not be available in real world scenario. However, the proposed algorithm requires 2-4 images for identity anonymization. OPOM has done evaluation on 500 identities and achieved 13.5% rank 1 and 19.9% rank 5 identification accuracy given the gallery has 1 image only. On the other hand, the proposed algorithm has done more comprehensive evaluation with 1,000 identities with multiple images in gallery and achieved lower rank 1 identification accuracy 5.38% and rank 5 identification accuracy 13.7% identification accuracy for 2 images in gallery. The proposed algorithm also outperforms with gallery size of 4 images, which showcase that the proposed algorithm is more effective compared to existing algorithms for multiple image anonymization.

3.5.3 Bounding Box Prediction

In the previous sections, we showcase the performance of the proposed algorithm for identity anonymization. To further showcase that the proposed algorithm can be extended for other tasks, we evaluated the proposed algorithm for bounding box prediction in face images. Correct bounding box prediction helps in localizing the face, which is a prerequisite for face recognition tasks in general. We performed the bounding box distance prediction on the CelebA-HQ dataset as this is the standard dataset used in the literature for comparison. The bounding box distance is performed using MtCNN (Zhang *et al.*, 2016) on both original and identity anonymized images, respectively. The identity anonymized images are generated using $\mathcal{L}_p$ norm. Table 3.8 shows the comparison of the results for bounding box distance using MtCNN. We achieved state-of-the-art (SOTA) results for the proposed algorithm, where the euclidean distance is 2.65. The second best results are reported by RIDDLE (Li *et al.*, 2023a), where the distance is 1.17 higher than the proposed algorithm. These results demonstrate the effectiveness of the proposed algorithm for preserving data utility while anonymizing images.

Table 3.8: Comparing the proposed algorithm with state-of-the-art identity anonymization algorithms on the CelebA-HQ dataset for bounding box distance prediction using MtCNN.

Anonymization	Bounding Box Distance
DeepPrivacy Hukkelås <i>et al.</i> (2019)	4.65
CIAGAN Maximov <i>et al.</i> (2020)	20.38
FIT Gu <i>et al.</i> (2020)	7.87
RIDDLE Li <i>et al.</i> (2023a)	3.82
FALCO Barattin <i>et al.</i> (2023)	7.88
Diff-Privacy He <i>et al.</i> (2024)	5.83
Proposed + $\mathcal{L}_p$	2.65

3.5.4 User Study

To evaluate the effectiveness of the proposed algorithm, we have performed a user study with 10 subjects (ages varying between 21 to 60 years, 4 males and 6 females) on 50 randomly chosen pairs of original and their corresponding perturbed images. These pairs consisted of both familiar and unfamiliar identities. The users were asked to perform two different tasks: 1) identify the original and perturbed image (generated through the PrIdentity algorithm) among the pairs and 2) identify if the original and their corresponding perturbed images belong to the same identity or not. Here, the first task measures the effectiveness of the proposed algorithm for data utility, while the second task measures the effectiveness for privacy.

For the first task, users could identify the original / perturbed images with an accuracy of 50%; interestingly, the errors are observed for both the classes, implying that the perturbations are not perceptible and the users are making a random guess. For the second task, the users were able to correctly identify all the pairs with a 100% matching rate since visual appearance is preserved after adding the learnt perturbations. It is important to note that the objective of the proposed algorithm is not to fool the human but to fool the face recognition models for privacy preservation. In several real-world applications like social media, it is important that the anonymized images are identifiable by humans. The results indicate that the proposed algorithm successfully incorporates privacy as well as data utility.

3.6 Identity Drift in Feature Space

Although the learned perturbation successfully removes the original identity information under an $\mathcal{L}_p$ -norm constraint, the anonymized embeddings may, in some cases, drift toward the feature regions of other real identities. This can lead to unintended identity associations or partial re-identification. Future work could investigate mechanisms to explicitly regularize the perturbation so that the anonymized representation moves toward unoccupied or synthetic “identity-free” subspaces. Incorporating additional objectives such as a triplet loss, where the anonymized embedding is enforced to be distant from both the original and all known identities, could help constrain this drift. Furthermore, adversarial or disentanglement-based regularization may be explored to ensure the anonymized embeddings remain distinct from any existing identity clusters in the feature space.

3.7 Summary

This research presented a novel privacy-preserving identity anonymization (PrIdentity) algorithm that suppresses identity information by learning and adding a perturbation using $\mathcal{L}_p$ norm in the given image. To showcase the effectiveness of our proposed algorithm, single and multiple image anonymization experiments are performed on the popular LFW, CelebA, CelebA-HQ datasets. It is observed that the proposed PrIdentity algorithm maintains a good balance between privacy and data utility. We have empirically shown that on learning the appropriate value for p , the proposed algorithm can successfully anonymize the identity of the given image and generalize well across unseen target deep models. The proposed PrIdentity algorithm can also be extended to a weighted scheme between privacy and data utility in future. This scheme will be highly beneficial for different organizations that outweigh one over the other. For instance, an organization would prefer data utility over privacy; however, a law-enforcement agency would opt for privacy. The proposed algorithm can be further extended for a dual objective, where the algorithm will be responsible for anonymizing identities as well as facial attributes.

Chapter 4

Low Quality Deepfake Detection via Unseen Artifacts

4.1 Introduction

With the significant increase in usage of social media platforms, millions of images and videos are shared by users on a daily basis. Nowadays, social media has become the primary source of information and is believed to circulate information faster than any other media. This encourages miscreants to indulge in generating and pushing fake content online for malicious gain. The recent advancements in AI-based systems have made the task of manipulating and generating high-quality fake videos with the desired content, much easier and more accessible. As a result, it has become very difficult for an ordinary person to distinguish between a real and a fake video with the naked eye. Recently, autoencoders and Generative Adversarial Networks (GANs) based techniques have become popular for generating high-quality synthetic videos which are also known as Deepfakes (Verdoliva, 2020). These techniques include face-swapping where the face of the person in the video is swapped with the target face and facial reenactment where the expression of one person is animated to another person. The ease of creating Deepfakes has also resulted in the misuse of AI technology. For example, the video of the former president of the US, Barack Obama posted by BuzzFeed on YouTube went viral in which he was seen to be speaking gibberish and yet, looked completely real (BuzzFeedVideo, 2018). Another example of misuse is an application called DeepNude, which went viral for automatically generating nude images. These examples showcase the need for a mechanism to identify manipulated media and prevent its misuse as it may pose a serious threat to the privacy and safety of individuals, as well as society at large.

Several researchers have developed algorithms to identify fake videos. In a real-world scenario, images and videos shared on social media platforms such as Facebook and WhatsApp undergo different kinds of compression to create medium or low-resolution videos, easy for transmission. Earlier research focused majorly on training and testing real and fake videos on the same compression factors. Kumar *et al.*

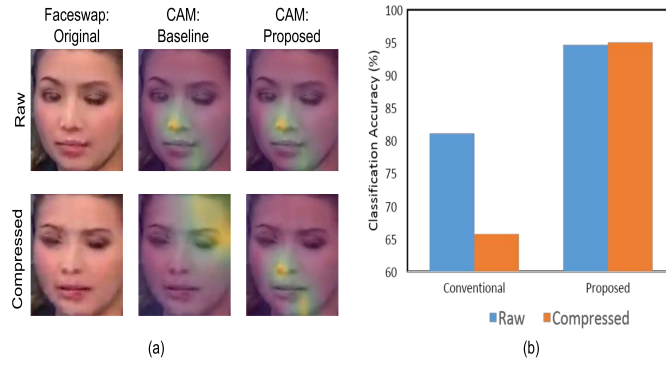


Figure 4.1: The Class Activation Maps (CAMs) shown in (a) depict that the conventional algorithms fail to focus properly after compression while the focus of the proposed algorithm stays consistent before and after compression. (b) shows that the proposed algorithm performs significantly better when trained on raw videos and tested on raw and compressed videos.

(2020) showed that the performance of the model trained on high-quality videos becomes worse when evaluated on low-quality or highly compressed videos. The low performance of the model in highly compressed videos is observed due to 1) loss or change of artifacts learned by the model in high-quality videos after compression and 2) new artifacts introduced because of compression. This makes the task of classification of real and fake videos difficult.

In general, the learning of a model is biased towards those artifacts or discriminative features which maximize the classification accuracy (Ilyas *et al.*, 2019). We hypothesize that these models learn features/artifacts that may get suppressed during compression. It is possible to improve the performance of such models by learning from other (more) artifacts as well. These unseen imperceptible artifacts can be potentially helpful in the detection of low-quality Deepfakes, as they reduce the chances of losing relevant features during high compression. In this research, we propose a novel algorithm for Deepfake video detection which is robust under different compression levels as illustrated in Figure 4.1. The proposed algorithm enforces the model to learn the unseen artifacts while maximizing the classification accuracy. In order to learn the unseen artifacts, we introduce a uniform visibility matrix v for a given dataset which is learned such that it represents the importance of different artifacts over the entire dataset. The visibility matrix is used to increase the importance of unseen imperceptible artifacts, which is then used for training the model.

In literature, the concept of Universal Perturbations (Moosavi-Dezfooli *et al.*, 2017) have been proposed, where the aim is to degrade the model performance through ad-

versarial attack. The universal adversarial perturbation introduces the artifacts in the images against the models that are already trained and are not robust, thereby results in degraded performance. Same is the case with low-quality Deepfakes where the compression algorithm introduces the compression artifacts, resulting in degraded performance of the trained models. On the contrary to the universal perturbation, the proposed algorithm improves the model performance on the compressed images by enhancing the unseen artifacts in the dataset using a visibility matrix. With the aid of this matrix, the proposed algorithm enhances the unseen features in the dataset to improve the model performance and enhance its robustness. The key highlights of this research are:

- A novel algorithm is proposed for Deepfake video detection by learning the unseen imperceptible artifacts.
- Introduced a uniform visibility matrix v which represents the importance of the pixels towards the model for a given dataset. This matrix is utilized to increase the importance of unseen imperceptible artifacts, which is then used to train the model.
- The experiments on intra- and inter-compression experiments conducted on models trained and tested for different quality of videos show that the proposed algorithm is robust under different compression factors.

4.2 Related Work

In the literature, researchers have proposed various algorithms for the generation and detection of manipulated images and videos. The traditional methods for generating fake images involve techniques such as copy-move forgery (Li *et al.*, 2014; Christlein *et al.*, 2012; Bianchi and Piva, 2012), splicing (Ng and Chang, 2004), and phylogeny (Dias *et al.*, 2011, 2013). With the advancements in machine learning and deep learning algorithms, the task of generating manipulated images and videos has become much easier. Dale *et al.* (2011) proposed a method to replace a face in the target video with the face in the source video. Their face replacement method uses a 3D multilinear model for facial tracking in both source and target videos and further warps the source video to the target video. Thies *et al.* (2016) proposed face2face approach for facial reenactment in real-time videos. They used a dense photometric consistency measure to capture facial expressions and performed reenactments by transferring deformation between the source and the target video. Kim *et al.* (2018) extended the work of Thies

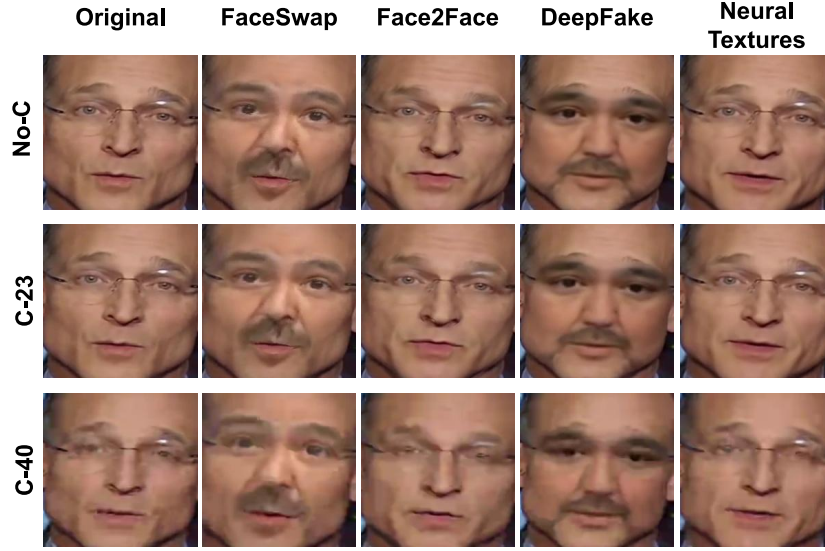


Figure 4.2: Samples from FaceForensics++ dataset manipulated using different techniques at easy, hard and no compression.

et al. (2016) by transferring full head 3D head position, head rotation, face expression, eye gaze, and eye blinking from source video to target video. More recently, GANs have been used to generate high-quality Deepfakes such as face swap (MarekKowalski (2016); DeepFakes (2017)), facial attribute manipulation (Choi *et al.*, 2018; Liu *et al.*, 2019; Shen and Liu, 2017), and self-reenactment (Thies *et al.*, 2016, 2019).

In order to obtain a better understanding of Deepfakes, researchers have created different datasets. FaceForesnics++ (Rossler *et al.*, 2019) dataset is proposed as an extension of FaceForensics (Rössler *et al.*, 2018). Four face manipulation techniques are applied to 1000 videos downloaded randomly from YouTube. The approaches used for generating these manipulations are Face2Face, FaceSwap, DeepFake, and NeuralTextures are shown in Figure 4.2. Li *et al.* (2019) created the Celeb-DF dataset which consists of 590 real videos and 5,639 Deepfake videos (with over 2 million frames). The real videos are taken from publicly available YouTube videos of celebrity interviews. The 59 celebrities in the videos have been selected considering diversity in terms of age, gender, and ethnicity. The videos also capture a wide variety of differences in lighting conditions, orientations, and size of subjects' faces. Zi *et al.* (2020) proposed a WildDeepfake dataset, which consists of 707 deep fake videos. This dataset is collected from the internet.

With continued improvement in the generation of fake videos, various algorithms have been proposed for detection (Rana *et al.*, 2022), (Wang *et al.*, 2024). Afchar *et al.*

(2018) proposed two concise deep learning networks designed to detect tampered faces in videos and showed how the mouth and eye regions played an important role in classification. Fernando *et al.* (2019) proposed a socio-cognitive approach to the problem of Deepfakes by using machine learning techniques inspired by how humans detect fake faces from real ones. They propose a Hierarchical Memory Network which utilizes previous knowledge of human faces as well as other visual cues, for classification. Li and Lyu (2018) use a deep learning approach to focus on face-warping artifacts induced by fake video generation methods. To train the network, they generate negative examples (fakes) using image processing techniques. Some popular algorithms use biological indicators such as eye blinking patterns to discriminate between real and fake media (Li *et al.*, 2018). Agarwal *et al.* (2019) utilized a person’s speaking patterns through facial expressions and movements for classification. Mittal *et al.* (2020) proposed a multimodal-based approach i.e. audio and video for the detection of Deepfakes. Chen *et al.* (2021b) proposed a mask-guided detection and reconstruction pipeline for defending the adversarial attacks to Deepfakes. Sun *et al.* (2021) proposed a robust and efficient framework named LRNet that captures temporal artifacts via geometric features for detecting Deepfakes. Zhao *et al.* (2021) proposed a multi-attentional Deepfake detection framework that enforces the network to learn from regions other than high-response regions. Wang and Dantcheva Wang and Dantcheva (2020) analyzed the state-of-the-art video CNNs to detect deep fake videos. They performed experiments on cross-manipulation techniques and concluded that the algorithms require a better understanding of manipulation techniques to detect unseen Deepfakes. Cao *et al.* (2022) proposed a reconstruction-classification learning (RECCE) framework To detect forged face images. The proposed reconstruction-based encoder-decoder framework models the genuine face distribution that is aware of the unknown forgery patterns. Liu *et al.* (2021) proposed a Spatial-Phase Shallow Learning (SPSL) method, which leverages the phase and spatial spectrum to capture artifacts for forgery detection.

Yan *et al.* (2023) proposed a disentanglement framework to address overfitting in Deepfake detection by separating forgery-irrelevant, method-specific, and common forgery features. Using multi-task learning and contrastive regularization, it enhances generalizability by leveraging only the common forgery features for detection. Yang *et al.* (2023) formulated Deepfake detection as a fine-grained classification task and proposed a multi-attentional network with spatial attention heads, textural feature enhance-

ment, and feature aggregation. Dong *et al.* (2023) identified implicit identity leakage as a key factor limiting the generalization of Deepfake detection models, where classifiers unintentionally learn identity-related features. To address this, it proposed the ID-unaware Deepfake detection model, which mitigates identity influence and enhances detection robustness. Le and Woo (2023) introduced QAD (Quality-Agnostic Deepfake Detection), a universal intra-model collaborative learning framework that enables effective detection across varying Deepfake quality levels. Ju *et al.* (2024) introduced novel loss functions to improve Deepfake detection fairness, addressing both scenarios with and without demographic information (e.g., race and gender). It aims to enhance detection accuracy while promoting fairness across diverse groups. Tan *et al.* (2024) revised CNN-based generator architectures to create a generalized representation of synthetic artifacts, revealing that upsampling operators produce both frequency-based and generalized forgery artifacts. It emphasizes the local interdependence of image pixels caused by upsampling in GAN or diffusion-generated images. Pellicer *et al.* (2024) proposed Prototype-based Unified Framework for Deepfake Detection (PUDD), which uses similarity-based video classification to identify Deepfakes and new classes by analyzing similarity drops. Shao *et al.* (2025) introduced DeepFake-Adapter, a parameter-efficient approach for Deepfake detection that adapts high-level semantics from pre-trained ViTs using lightweight dual-level adapter modules. It combines globally-aware and locally-aware adapters to capture both global and local forgery cues, improving detection by leveraging high-level features and low-level Deepfake patterns.

Based on the literature review, it is observed that the majority of the algorithms focus on the detection of Deepfakes assuming the test input would be of the same quality on which the model is trained. Kumar *et al.* (2020) showed that the performance of the model trained on high-quality videos becomes worse when evaluated on low-quality or highly compressed videos. The reason behind this is the loss or change of artifacts because of compression showcasing the urgent need for an algorithm that is robust under different compression factors.

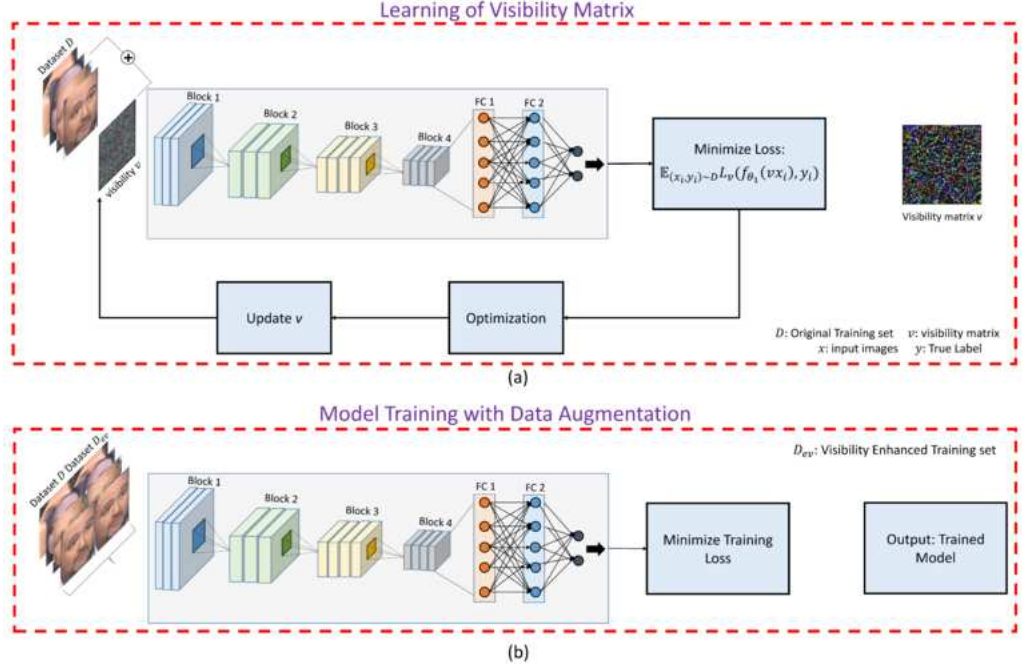


Figure 4.3: Illustrating the steps involved in the proposed algorithm. (a) The visibility matrix v is learned over a randomly sampled batch of images to enhance the visibility of unseen artifacts. (b) A new dataset with enhanced visibility of artifacts is created and augmented with the original dataset. The model is re-trained over the new dataset. Steps (a) and (b) are performed iteratively.

4.3 Proposed Algorithm

Various artifacts are introduced while generating high-quality fake images/videos. In general, the artifacts that are sufficient for performing confident classification between real and fake are utilized by the model. This results in limited learning of the model as many other artifacts are left unseen by the model. During compression, the artifacts that are already learned by the model may be lost due to quality degradation which is the primary reason for the low performance of existing approaches (Kumar *et al.*, 2020). Hence, it is important to learn more artifacts (unseen imperceptible) apart from the ones that have been previously learned. To force the model to learn more (unseen imperceptible) artifacts, we propose a novel algorithm that learns artifacts that persist even after compression. This leads to a robust, generalizable model that performs well even for lower-quality images/videos. To compute and enhance the visibility/importance of the artifacts, we introduce a uniform visibility matrix v which represents the importance of the pixels in any given image of the dataset. The aim is to enhance the visibility of the unseen imperceptible artifacts present in the input images for better classification and Figure 4.3 shows the steps involved in the proposed algorithm.

Let D be the dataset with m number of high-quality frames of different videos.

$$D = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\} \quad (4.1)$$

Here, x_i represents a video frame and y_i represents the label of a video frame encoded in one hot form. y_i can be either real or fake i.e., 0 or 1. The proposed algorithm is categorized into two stages: 1) Learning visibility matrix for unseen imperceptible artifacts and 2) Training with unseen artifacts. The details of both stages are discussed below.

4.3.1 Learning Visibility Matrix for Unseen Imperceptible Artifacts

Let v be the uniform visibility matrix for a given dataset D . Each element in the visibility matrix v is initialized with one. The size of the visibility matrix v is the same as the size of an input image x_i . Let D_{ev} be the dataset with enhanced visibility of unseen imperceptible artifacts generated by multiplying v with dataset D .

$$D_{ev} = \{(vx_1, y_1), (vx_2, y_2), \dots, (vx_m, y_m)\} \quad (4.2)$$

where, vx_i is the element-wise multiplication of v and x_i . As mentioned above, all the elements of the visibility matrix are initialized with one. Hence, in the initial step, $D_{ev} = D$. In order to learn the visibility matrix v for any model, the first step is to train the model with dataset D for a few epochs. Let f_{θ_1} is the trained model on dataset D with parameters θ_1 . For a given input image x_i , the output of the model is written as:

$$p(y_i|x_i) = f_{\theta_1}(x_i) \quad (4.3)$$

where, $p(y_i|x_i)$ represents the output predicted score for a given input image x_i . The model targets to minimize the distance between the predicted score and the true label. Since our aim is to learn the visibility matrix v , therefore the distance between true class and predicted score is minimized by freezing the model f_{θ_1} and updating the matrix v only. Mathematically, it is written as:

$$\min_v \mathbb{E}_{(x_i, y_i) \sim D} L_v(f_{\theta_1}(vx_i), y_i) \quad (4.4)$$

where $L_v(\cdot)$ represents the function to compute the distance between the predicted class score and true class. Since the elements in visibility matrix v are initialized with one, minimizing the above equation would result in the visibility matrix v with values less than or equal to one, and greater than one. The values in v greater than one corresponds to the combination of pixels the model has learned as an artifact. On the other hand, the values less than or equal to one corresponds to pixels the model is enforcing towards zero. Therefore, the term vx_i in the above loss function results in the suppression of a few pixels of x_i towards zero for the values in v less than one and scaling of the rest of the pixels towards higher values for values in v greater than one. It depicts that the model is primarily learning the artifacts from a few pixels only corresponding to values that are greater than one in the matrix v while enforcing other pixels towards zero.

In Eq. 4.4, the function $L_v(\cdot)$ is updated to minimize the distance between predicted class scores and true class.

$$\min_v \mathbb{E}_{(x_i, y_i) \sim \mathbf{D}} \max(0, 1 - f_{\theta_1}(vx_i)y_i) \quad (4.5)$$

where, the product of $f_{\theta_1}(vx_i)$ and y_i outputs the true class score. Using Eq. 4.3, the above equation is updated as:

$$\min_v \mathbb{E}_{(x_i, y_i) \sim \mathbf{D}} \max(0, 1 - p(y_i|vx_i)y_i) \quad (4.6)$$

In this research, the visibility matrix v is learned over randomly sampled multiple batches of images. Let n be the number of images in a particular batch b . Thus, the above equation is written as:

$$\min_v \sum_{i=1}^n \max(0, 1 - p(y_i|vx_i)y_i) \quad (4.7)$$

Once the visibility matrix v is learned, the next task is to enhance the unseen artifacts in the input images using v and further train the model. It is our hypothesis that the unseen artifacts are present in the pixel regions which are suppressed by the model. In other words, the unseen artifacts lie in the region corresponding to values less than one in v . Hence, to enhance the unseen artifacts, we want to suppress pixels corresponding to v whose values are greater than one and upscale those pixels whose values are less

Algorithm 1: Model training algorithm

Input: Trained model f_{θ_1} , Dataset D **Output:** Trained model f_{θ_2} , uniform visibility matrix v

Parameters: Number of Epochs: e , Number of iterations: t , Number of Steps: s **Initialize:**

$D_{ev} = D, f_{\theta_2} = f_{\theta_1}$ **for Steps = 1 to s do**

$v = \text{compute_visibility_matrix}(D_{ev}, f_{\theta_2})$ $v = \text{Clip}\{-\lambda(v - 1) + 1\}$

$D_{ev} = (vx_1, y_1), (vx_2, y_2), \dots, (vx_m, y_m)$ Augmented Dataset $D_{aug} = \{D_{ev}, D\}$

for Epochs = 1 to e do

for every batch of size a do

 Forward pass $f_{\theta_2}(D), f_{\theta_2}(D_{ev})$ **Calculate loss:**

$-y_i(\log(p(y_i|vx_i)) + \log(p(y_i|x_i)))$ Calculate gradient and update θ_2

end

end

end

Function *compute_visibility_matrix*

Initialize: visibility matrix $v = 1$, Freeze model f_{θ_2}

for iterations = 1 to t do

 Randomly select batch b of size n Forward pass $f_{\theta_2}(D_{ev})$ **Calculate loss:** =

$\sum_{i=1}^n \max(0, 1 - p(y_i|vx_i)y_i)$ Calculate gradient and update v

end

end

than one in v . To achieve this, we update v using the following function:

$$v = \text{Clip}\{-\lambda(v - 1) + 1\} \quad (4.8)$$

where, λ is a hyperparameter. The term $v - 1$ outputs the difference between the initial visibility matrix which is initialized to one and the learned visibility matrix v . The output difference will be centered around zero. Multiplying this output term with λ strengthens the visibility and multiplying with -1 suppresses the enhanced visibility. On adding the output with 1 shift the difference to be centered around 1. For enhancement of unseen imperceptible artifacts using updated visibility matrix v , Equation 4.2 is employed.

4.3.2 Training with Unseen Imperceptible artifacts

Once the dataset D_{ev} with enhanced visibility of unseen artifacts is obtained, the next step is to train the model f_{θ_1} with D_{ev} . The model f_{θ_1} is pre-trained on dataset D with parameters θ_1 and the cross entropy loss is minimized.

$$\min_{\theta_2} \mathbb{E}_{(vx_i, y_i) \sim D_{ev}} - y_i \log(f_{\theta_2}(vx_i)) \quad (4.9)$$

where, f_{θ_2} is the updated model with parameters θ_2 . Here, y_i is in one hot form and only one element in the y_i vector will be non-zero. Therefore, the cross-entropy loss is minimized only for true class-predicted scores. Using equation 4.3, the above equation is also written as:

$$\min_{\theta_2} \mathbb{E}_{(vx_i, y_i) \sim \mathbf{D}_{ev}} - y_i \log(p(y_i | vx_i)) \quad (4.10)$$

However, it is important that while learning model f_{θ_2} , the artifacts learned by the model f_{θ_1} should not be forgotten. Therefore, the model f_{θ_2} is learned over D_{ev} along with the augmented original dataset D as well to retain the artifacts learned by f_{θ_1} . Thus, the above equation is updated as:

$$\min_{\theta_2} \mathbb{E}_{(x_i, y_i) \sim \mathbf{D}} \mathbb{E}_{(vx_i, y_i) \sim \mathbf{D}_{ev}} - y_i (\log(p(y_i | vx_i)) + \log(p(y_i | x_i))) \quad (4.11)$$

It is important to note that stages 2 and 3 are performed iteratively. The training procedure for learning v and f_{θ_2} iteratively is shown in Algorithm 1.

4.4 Datasets and Experimental Details

The experiments are performed on the FaceForensics Rössler *et al.* (2018), FaceForensics++ Rossler *et al.* (2019), and Celeb-DF Li *et al.* (2020) datasets. The details of all the datasets and their protocol are discussed below.

FaceForensics (Rössler *et al.*, 2018): The facial expression manipulation is performed using the Face2Face approach Thies *et al.* (2016). This dataset contains 1004 original and altered videos with 704 videos for training, 150 videos for validation, and 150 videos for testing. For each video, 10 frames are randomly sampled from both original and altered videos to evaluate the performance using the standard protocol defined by Rössler *et al.* (2018).

FaceForensics++ (Rossler *et al.*, 2019): The dataset consists of 1,000 original and altered videos with 720 videos for training, 140 videos for validation, and 140 videos for testing. These videos are downloaded randomly from YouTube. In this dataset, three different face manipulation techniques are applied namely, Face2Face (Thies *et al.*, 2016), FaceSwap (MarekKowalski, 2016), and DeepFake (DeepFakes, 2017). Each

Table 4.1: Details of the experiments to showcase the effectiveness of the proposed algorithm.

Experiment	Datasets	Models Trained on	Models Evaluated on
Intra- Compression	FaceForensics, FaceForensics++ Celeb-DF	c-40	c-40
Inter- Compression	FaceForensics, FaceForensics++ Celeb-DF	c-0	c-23, c-40
		c-23	c-0, c-40
		c-40	c-0, c-23

video is compressed with three different compression factors, similar to the FaceForensics dataset. The standard pre-defined protocol is followed for the experiments.

Celeb-DF-v2 (Li *et al.*, 2020): The dataset consists of 590 original videos, 5639 altered videos, and additional 300 original YouTube videos. For training, the dataset is split into 6,011 train videos and 518 test videos using the standard protocol. Each video in this dataset is compressed with three different compression factors as present in the FaceForensics and FaceForensics++ datasets.

It is important to note that all the training and evaluations are performed on the FaceForensics, FaceForensics++, and Celeb-DF datasets corresponding to c-0, c-23, and c-40 compressions. The c-0 refers to no-compression which are the high-quality videos without any compression. “c-23” involves light compression (constant rate quantization parameter equal to 23) while “C-40” involves hard compression resulting in a low-quality video (LQ) (constant rate quantization parameter equal to 40). These are the standard compression factors used in Rössler *et al.* (2018); Rossler *et al.* (2019); Li *et al.* (2020) for the training and evaluation.

To evaluate the performance of the proposed algorithm, we have performed two experiments, namely: 1) **Intra Compression** and 2) **Inter Compression**. Table 4.1 shows the details of both experiments. The first experiment represents the scenario where the model trained with specific quality/compression of videos is evaluated for videos with the same quality/compression factor. For intra-compression, Kumar *et al.* (2020) and Rossler *et al.* (2019) have demonstrated that very high performance has been achieved for c-0 (no compression) and c-23 (light-compression) videos. However, c-40 (hard

Table 4.2: Classification accuracy (%) on FaceForensics and FaceForensics++ dataset for intra compression experiment at c-40 (hard compression).

FaceForensics		FaceForensics++			
Methods	Face2Face	Methods	DeepFake	Face2Face	FaceSwap
Bayar and Stamm (2016)	73.63	Bayar and Stamm (2016)	77.30	76.83	72.38
Han <i>et al.</i> (2017b)	86.83	Rahmouni <i>et al.</i> (2017)	62.33	67.08	62.59
Afchar <i>et al.</i> (2018)	83.20	Afchar <i>et al.</i> (2018)	89.52	84.44	83.56
Kumar <i>et al.</i> (2020)	91.20	Sabir <i>et al.</i> (2019)	96.90	94.35	96.30
Proposed + XceptionNet	94.00	SSTNet (Wu <i>et al.</i> (2020))	95.33	90.48	94.09
Proposed + ResNet50	94.33	WildDF (Zi <i>et al.</i> (2020))	90.11	-	-
		I3D (Wang and Dantcheva (2020))	95.13	90.27	92.25
		SPSL (Liu <i>et al.</i> (2021))	93.48	86.02	92.26
		Proposed + XceptionNet	95.71	90.35	94.29
		Proposed + ResNet50	97.14	91.79	93.57

compression) is still a challenging task. Therefore, in this research, we particularly focus on c-40 compression. In the second experiment, the model is trained with specific quality/compression and evaluated for other compression factors. This experiment is performed for all three compression factors i.e., c-0 (no compression), c-23 (light compression), and c-40 (hard compression) videos. On the Celeb-DF dataset, an additional experiment *cross-dataset + inter-compression* is performed to showcase the effectiveness of the proposed algorithm by comparing the results with existing algorithms.

For both the experiments, XceptionNet (Chollet, 2017) and ResNet50 (He *et al.*, 2016) models, pre-trained on the ImageNet (Krizhevsky *et al.*, 2012) dataset, are used as the base model. These models are chosen for the experiments to follow the standard protocols as described in (Rössler *et al.*, 2018; Rossler *et al.*, 2019). In the first experiment, the performance of the proposed algorithm is evaluated on all three datasets and the results are compared with existing algorithms on both FaceForensics and FaceForensics++ datasets. In the second experiment, the performance of the proposed and existing algorithms are compared on the FaceForensics and Celeb-DF datasets. For the FaceForensics++ dataset, the performance is compared with fine-tuned models since no existing approaches have shown their results for inter-compression on this dataset.

4.4.1 Implementation Details

All the experiments are performed on a Intel Xeon processor based workstation with 128GB RAM and Nvidia GeForce RTX 2080Ti GPU. The proposed algorithm is im-

plemented in Tensorflow 2.0. Images are resized to 128×128 resolution. For both the experiments, XceptionNet (Chollet, 2017) and ResNet50 (He *et al.*, 2016) pre-trained models are used as the base model. For fine-tuning of base models ResNet50 and XceptionNet, two fully connected layers of dimension 512 followed by a softmax layer are added to the pre-trained model and fine-tuned for 20 epochs. The learning rate is set to 0.0001 and after 10 epochs, the learning is reduced to 0.00001. A batch size of 32 is chosen. The hyperparameters for all the experiments are optimized using a grid search approach. The parameters used for intra- and inter-compression experiments are discussed below.

Intra-Compression: Initially, the models are fine-tuned for 20 epochs with a batch size of 32. The learning rate is set to 0.0001 and reduced to 0.00001 after 10 epochs. These models are treated as pre-trained models for the proposed algorithm, which are further trained by setting the number of steps to 6 and epochs to 4. The learning rate is set to 0.0001 and reduced to 0.00001 after the second epoch. For computing visibility matrix v , the number of iterations is set to 3 and a batch size of 100 is used. The hyperparameter λ_{init} is initialized with 1.3 and increased to $\lambda = \lambda_{init} \times steps$. The clipping value of the visibility matrix is set to 2.

Inter-Compression: Model training parameters are the same as those for the Intra-Compression experiment. For computing the visibility matrix v , the number of iterations is set to 10 and a batch size of 50 is used. The hyperparameter λ_{init} is initialized with 10 and increased to $\lambda = \lambda_{init} \times steps$. The clipping value of the visibility matrix is set to 2.

Computational Runtime of the Proposed Algorithm: As mentioned in the section 4.3, the proposed algorithm is categorized into two stages: 1) Learning visibility matrix for unseen imperceptible artifacts and 2) Training with unseen artifacts. When the model is trained with the dataset $\mathbf{D}$, each epoch takes 56 seconds. In the first stage, the visibility matrix is learned to enhance the visibility of unseen imperceptible artifacts. The total time for learning the visibility matrix is 10 seconds. In the next stage, the model is trained with the original dataset $\mathbf{D}$ and the dataset $\mathbf{D}_e v$ with enhanced visibility of unseen imperceptible artifacts. In this stage, each epoch takes 110 seconds as the size of the dataset gets doubled. In the training process, initially the model is trained for few epochs, while stage one and two are repeated iteratively. The total time spent

Table 4.3: Classification accuracy (%) on c-0, c-23, and c-40 videos for inter compression experiment on the FaceForensics dataset.

Train	Test	Kumar et al. Kumar <i>et al.</i> (2020)	Proposed + XceptionNet	Proposed + ResNet50
c-0	c-23	58.26	92.00	92.33
	c-40	52.66	70.00	78.00
	Overall	55.46	81.00	85.16
c-23	c-0	99.56	96.00	96.33
	c-40	55.43	88.00	83.67
	Overall	77.49	92.00	90.00
c-40	c-0	96.73	96.00	95.00
	c-23	95.76	96.00	95.00
	Overall	96.24	96.00	95.00

on the training of an intra-compression model is 62 mins 58 seconds while the training time for an inter-compression model is 63 mins 40 seconds, respectively.

4.5 Experimental Results and Analysis

The experimental protocol for both intra-compression and inter-compression experiments is summarized in Section 4.4. The detailed results of the experiments and their analysis are discussed in the following subsections.

4.5.1 Intra-Compression

In this experiment, we trained various models on c-40 compressed images/videos and evaluated them on the same compression. In general, it is observed that the detection accuracy on c-40 (highly compressed) videos is comparatively lower than the detection accuracy on c-0 and c-23 videos Rossler *et al.* (2019); Kumar *et al.* (2020). The proposed algorithm yields an accuracy of 94% and 94.33% with XceptionNet and ResNet50 as the base models respectively on the FaceForensics dataset. Similarly, the proposed algorithm yields an accuracy of 88.88% and 82.43% respectively on the Celeb-DF dataset. In the case of the FaceForensics++ dataset, the classification accuracy obtained using

Table 4.4: Classification accuracy (%) on c-23, and c-40 videos when the model is trained on c-0 videos for inter-compression experiment on the FaceForensics++ dataset. The classification accuracy corresponds to Deepfake (DF), Face2Face (F2F), and FaceSwap (FS) alterations with XceptionNet and ResNet50 as a base model.

Manipulation Type	XceptionNet				ResNet-50			
	Fine-tuned		Proposed		Fine-tuned		Proposed	
	c-23	c-40	c-23	c-40	c-23	c-40	c-23	c-40
DeepFake	81.07	65.71	94.64	95.00	91.43	69.29	93.57	93.21
Face2Face	87.14	51.07	94.64	73.21	94.29	55.71	93.57	77.50
FaceSwap	92.50	58.57	95.00	91.79	94.29	68.21	94.29	92.14
Mean	72.68		90.71		78.87		90.71	

Table 4.5: Classification accuracy (%) on c-0, and c-40 videos when the model is trained on c-23 videos for inter-compression experiment on the FaceForensics++ dataset. The classification accuracy corresponds to Deepfake (DF), Face2Face (F2F), and FaceSwap (FS) alterations with XceptionNet and ResNet50 as a base model.

Manipulation Type	XceptionNet				ResNet-50			
	Fine-tuned		Proposed		Fine-tuned		Proposed	
	c-0	c-40	c-0	c-40	c-0	c-40	c-0	c-40
DeepFake	97.50	87.50	94.29	95.36	97.14	92.50	94.29	93.93
Face2Face	97.14	67.86	96.43	83.21	98.21	69.29	95.71	88.57
FaceSwap	98.57	85.36	94.29	91.07	97.86	86.07	96.79	92.14
Mean	88.99		92.44		90.18		93.57	

the proposed algorithm varies from 90.35% to 97.14% for different alterations. The higher performances on these datasets showcase the effectiveness of the proposed algorithm when models are trained and evaluated on highly compressed images/videos.

We have also compared the performance of the proposed algorithm with various existing algorithms on both FaceForensics and FaceForensics++ datasets, respectively. Existing algorithms primarily focus on designing different deep model architectures and feature extraction techniques for Deepfake detection. For instance, Sabir *et al.* (2019) explored different combinations of recurrent convolutional models to exploit temporal artifacts for Deepfake detection. Wu *et al.* (2020) proposed an SSTNet framework, which extracts spatial and temporal features for face manipulation detection. Liu *et al.*

Table 4.6: Classification accuracy (%) on c-0, and c-23 videos when the model is trained on c-40 videos for inter-compression experiment on the FaceForensics++ dataset. The classification accuracy corresponds to Deepfake (DF), Face2Face (F2F), and FaceSwap (FS) alterations with XceptionNet and ResNet50 as a base model.

Manipulation Type	XceptionNet				ResNet-50			
	Fine-tuned		Proposed		Fine-tuned		Proposed	
	c-0	c-23	c-0	c-23	c-0	c-23	c-0	c-23
DeepFake	91.79	92.14	94.64	94.28	96.79	95.71	96.07	95.00
Face2Face	92.86	90.71	92.14	91.07	92.86	91.07	92.86	91.78
FaceSwap	94.29	89.64	93.21	91.07	94.64	91.07	93.21	93.57
Mean	91.91		92.74		93.69		93.75	

(2021) proposed a method to learn artifacts from the phase and spatial spectrum for face forgery detection. The comparison of the results with existing algorithms is shown in Table 4.2. It is observed that the proposed algorithm achieves state-of-the-art results among the top-performing existing algorithms, on the FaceForensics dataset for both the base models. On the FaceForensics++ dataset, the proposed algorithm achieves state-of-the-art performance for Deepfake alteration and the second-best performance for Face2Face alteration. The results indicate that the proposed algorithm is able to enhance and learn the unseen/suppressed artifacts in low-quality videos which shows the effectiveness of the algorithm in detecting low-quality manipulated videos.

In general, several artifacts may be introduced while generating fake videos. Only some of these artifacts are visually perceptible and interpretable by humans Ilyas *et al.* (2019). Figure 4.4(a) shows samples for Face2Face alteration with visually perceptible artifacts highlighted around the face contour. It is important to mention here that deep models do not necessarily learn visually perceptible and human interpretable features for detection, and might learn other features (non-interpretable) as well Ilyas *et al.* (2019). Figure 4.4(b) shows the visualization of the visibility matrices learned corresponding to the XceptionNet model at two different steps. In step 1, the model learns artifacts present around the face contour along with other artifacts. However, in step 2, these artifacts are suppressed and other unseen artifacts are enhanced by the model.

Table 4.7: Mean NIQE quality scores on the FaceForensics++ dataset corresponding to c-0, c-23, and c-40 compressions, respectively. Lower scores represent better quality.

	Original	Face2Face	Deepfake	FaceSwap
c-0	40.39	41.43	43.22	39.86
c-23	40.89	41.51	43.31	40.97
c-40	45.29	46.24	47.25	45.95

4.5.2 Inter Compression

This experiment addresses the scenario where the model is trained on specific quality and evaluated on other quality images/videos. It is performed for all three compression factors. In order to measure the performance of the models, individual and overall classification accuracy metrics are used. Individual classification accuracy represents the accuracy obtained on a specific compression while the overall classification accuracy represents the mean of the individual classification accuracies obtained on all three compressions corresponding to the model trained on a specific quality/compression. We will first discuss the results on the FaceForensics dataset followed by the results on the FaceForensics++ and Celeb-DF datasets.

Performance on FaceForensics: Table 4.3 shows the classification accuracy obtained using the proposed algorithm. Generally, it is observed that models trained on low-compression images/videos do not perform well on high-compression images/videos Kumar *et al.* (2020). This is due to the suppression of artifacts after compression. It is observed from Table 4.3 that the proposed algorithm with ResNet50 gives classification accuracy between 78% and 92% when the model is trained on low-compression images and tested for high-compression images. On comparing the overall classification accuracy, the proposed algorithm outperforms the existing algorithm Kumar *et al.* (2020) when the models are trained on c-0 and c-23 videos while achieving comparable performance for models trained on c-40 videos. For instance, the overall classification accuracy obtained corresponding to the model trained on c-0 videos using the proposed algorithm with XceptionNet as the base model is 81%, which is 25.54% higher than Kumar *et al.* (2020). A similar observation is made for c-40 videos when ResNet50 is used as the base model and trained on c-0 videos.

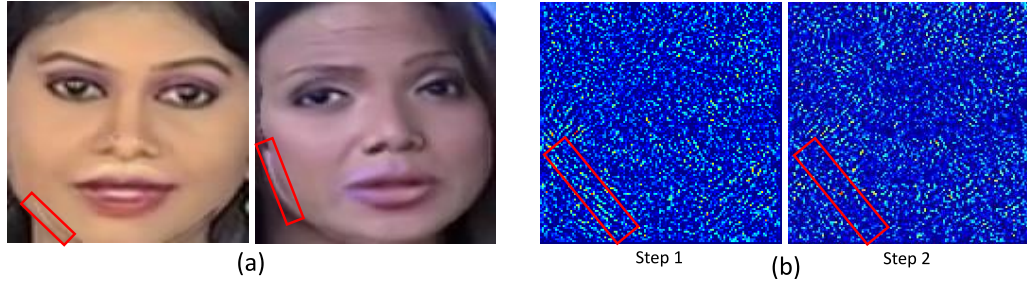


Figure 4.4: Samples with highlighted artifacts corresponding to Face2Face alteration on the FaceForensics++ dataset. (a) shows the samples with visually imperceptible artifacts in the bounding box. (b) shows the visualization of the visibility matrices in the jet colormap at step 1 and step 2.

As mentioned earlier, the aim is to enhance the performance of the model for low-quality videos when the model is trained on high-quality videos. The high performance of the proposed algorithm on low-quality videos demonstrates that the proposed algorithm is able to learn the unseen artifacts from high-quality/no-compression videos which are also present in low-quality/hard-compression videos.

Performance on FaceForensics++: Tables 4.4, 4.5, and 4.6 show the classification accuracy obtained using the proposed algorithm for Deepfake, Face2Face, and FaceSwap alterations with XceptionNet and ResNet50 as the base models. The results are compared with the models fine-tuned for the same. For the Deepfake alteration, we observe that the proposed algorithm performs exceptionally well when trained on lower compression and tested on higher compression 4.5, and 4.6. The proposed algorithm outperforms the fine-tuned model by approximately 30% when trained with c-0 images/videos and evaluated on c-40 images/videos with XceptionNet as the base model. While evaluating the performance for the Face2Face alteration, it is observed that the fine-tuned model gives close to random performance when trained with c-0 and evaluated on c-40 images/videos, respectively. This showcases that the artifacts learned by the fine-tuned models on c-0 images/videos get lost during compression, leading to poor performance. However, the proposed algorithm performs significantly better by 22.14% and 21.79% when XceptionNet and ResNet50 are used as the base models. This is because the proposed algorithm learns artifacts that persist even after compression. This exhibits the applicability of the proposed algorithm in real-world scenarios.

On comparing the overall performance of the proposed algorithm with the fine-tuned models from Table 4.4, 4.5, and 4.6, it is observed that the proposed algorithm outperforms the fine-tuned models across all alterations with XceptionNet and

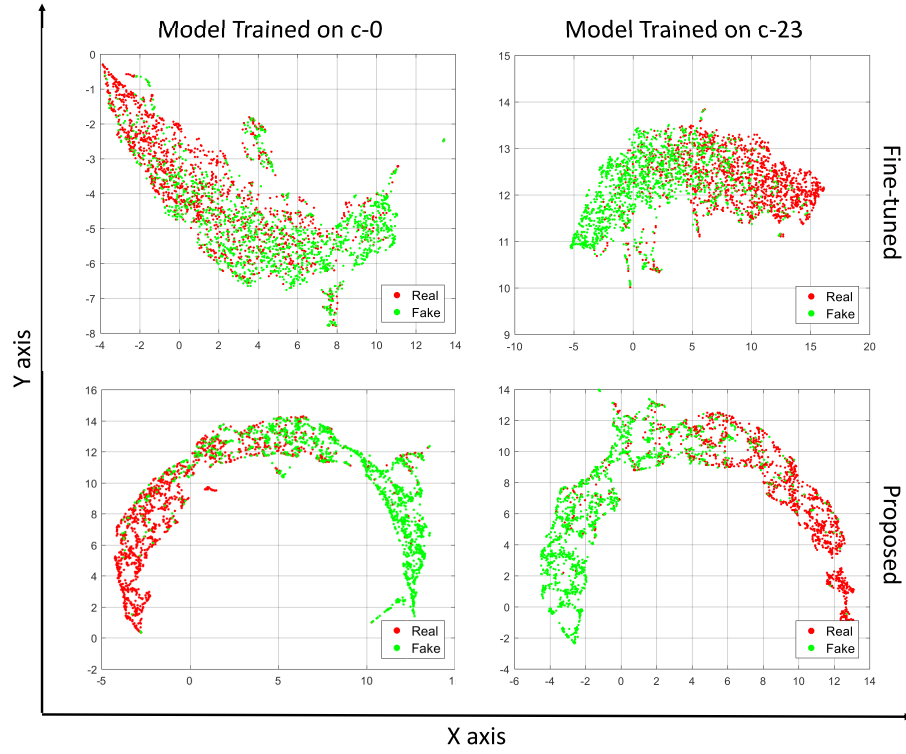


Figure 4.5: UMAPs depicting the separation between real and fake images in c-40 videos. The fine-tuned and proposed models are trained on c-0 and c-23 individually.

ResNet50 as the base models. For instance, the overall classification accuracy obtained using the proposed algorithm for the models trained on c-0 videos, across all the alterations and compression factors is 90.71%, which is 11.84% higher than the fine-tuned models as shown in Table 4.4. A significant performance enhancement is observed using the proposed algorithm which showcases its efficacy. With ResNet50 as the base model, the fine-tuned models give an overall classification accuracy of 78.87%, 90.18%, and 93.69% respectively on c-0, c-23, and c-40 videos, and the proposed algorithm

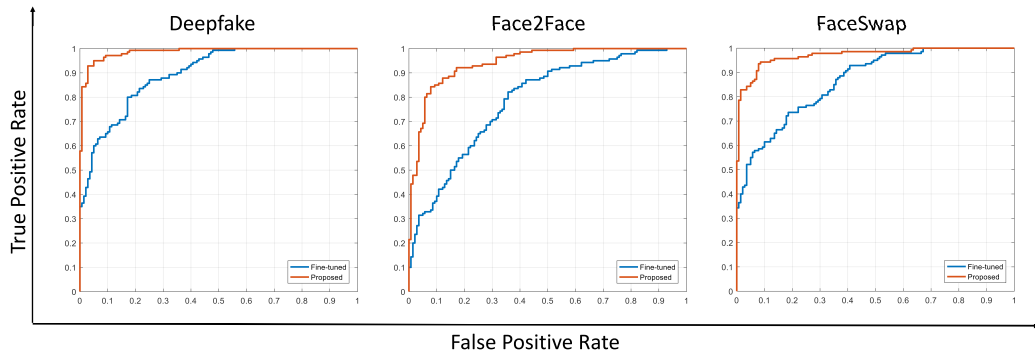


Figure 4.6: ROC curves comparing the performance of the fine-tuned model and the proposed algorithm on c-40 videos. The models are trained using c-0 videos with XceptionNet as the base model.

Table 4.8: Classification accuracy (%) on c-0, c-23, and c-40 videos corresponding to the proposed algorithm for the inter-compression experiment on the Celeb-DF dataset.

Train on	Test on	Proposed + XceptionNet	Proposed + ResNet50
c-0	c-23	94.20	93.82
	c-40	83.01	85.90
	Overall	86.60	89.86
c-23	c-0	93.82	93.62
	c-40	88.03	87.45
	Overall	90.92	90.53
c-40	c-0	87.64	88.03
	c-23	86.67	87.06
	Overall	87.83	87.54

gives an overall classification accuracy of 90.71%, 93.57%, and 93.75%, respectively. The high performance of the proposed algorithm for both XceptionNet and ResNet50 showcases the generalizability of the proposed algorithm across different deep models, as well. The performance of the proposed algorithm is further analyzed using different visual representations and compared with the fine-tuned models.

Figure 4.5 shows the Uniform Manifold Approximation and Projection (UMAP) visualizations of the output feature representations obtained using the proposed algorithm and the fine-tuned models for Face2Face alteration. These visualizations correspond to c-40 videos when the model is trained on c-0 videos and c-23 videos. The visualization shows that the proposed algorithm yields better results by reducing the overlap between original and altered representations. Figure 4.6 shows the ROC curves obtained using the proposed algorithm and fine-tuned models for all three alterations, with XceptionNet as the base model. These ROC curves correspond to c-40 videos when the model is trained on c-0 videos. It is observed from Figure 4.6 that the proposed algorithm enhances the performance of the model for all three alterations which showcases the generalizability of the proposed algorithm across different alterations. From these ROC curves, it can also be inferred that Face2Face manipulations are more sensitive to compression artifacts compared to Deepfake and FaceSwap manipulations.

To estimate the quality of the dataset used for training and evaluation of the proposed



Figure 4.7: Class Activation Maps for some samples that are misclassified as real by the fine-tuned model but correctly classified by the proposed model. (a) is original image classified as Deepfake while (b) and © are Deepfake images classified as original.

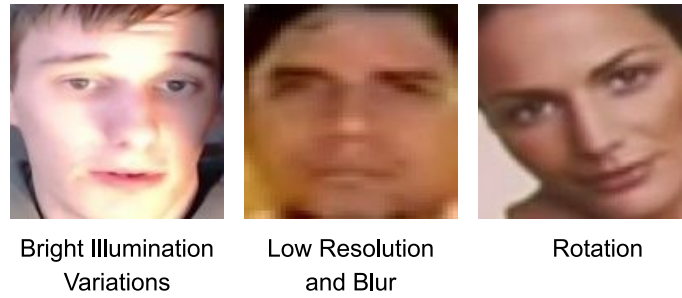


Figure 4.8: Samples misclassified by the proposed algorithm on the FaceForensics++ dataset.

algorithm, we computed the mean NIQE quality Mittal *et al.* (2012) on the FaceForensics++ dataset for different alteration methods corresponding to c-0, c-23, and c-40 compressions, respectively. The NIQE quality scores are shown in Table 4.7. From Table 4.7, we have observed the following:

- The quality degradation of the images/videos is minimal for the original and all alteration methods when comparing the quality scores of c-23 against c-0.
- There is a significant drop in the quality scores for the c-40 compression, which is the primary root cause for the degradation of the model’s performance trained on either c-0 or c-40 compression images.
- The mean quality scores of the Face2Face and Deepfake methods are lower than the original. On the other hand, the quality scores of the faceswap method are



Figure 4.9: Illustration of Deepfakes publicly circulated on different social media platforms. Typically, on social media platforms, videos undergo different levels of compression. The proposed algorithm successfully identifies the above in-the-wild samples to be fake.

comparable to the original for all the compressions.

The class activation maps obtained using GradCAM Selvaraju *et al.* (2017) of some altered frames misclassified as original frames using the fine-tuned model that are correctly classified using the proposed algorithm are shown in Figure 4.7. The result corresponds to the model trained on c-0 videos and evaluated on c-40 videos for FaceSwap alteration using ResNet50 as the base model. It is observed that the proposed algorithm focuses on the nose and lips to extract distinguishable features for differentiating original and altered frames. From all these analyses, it is clear that the proposed algorithm is generalizable across different models as well as different alterations. This shows the applicability of the proposed algorithm for detecting altered videos.

We also analyze the proposed algorithm on some failure cases illustrated in Figure 4.8. Based on the visual inspection, we found that the images impacted by the bright illumination variations, pose, rotation, low resolution, and blur are misclassified by the proposed algorithm. These samples are out-of-distribution for the proposed algorithm.

Performance on Celeb-DF: We next evaluate the effectiveness of the proposed algo-

Table 4.9: AUC scores (%) on different compression factors on the Celeb-DF dataset. The scores are reported for cross dataset+inter-compression experiment. The model is trained on the FaceForensics++ dataset and evaluated on the Celeb-DF dataset.

Methods	c-0	c-23	c-40
Two-stream Zhou <i>et al.</i> (2017)	53.80	-	-
MesoInception4 Afchar <i>et al.</i> (2018)	53.60	-	-
Multi-task Nguyen <i>et al.</i> (2019a)	54.30	-	-
Meso4 Afchar <i>et al.</i> (2018)	54.80	-	-
FWA Li and Lyu (2018)	56.90	54.60	52.20
Capsule Nguyen <i>et al.</i> (2019b)	57.50	-	-
EfficientNet-B4 Tan and Le (2019)	64.29	-	-
XceptionNet-c23 Li <i>et al.</i> (2020)	65.30	65.50	52.50
XceptionNet-c40 Li <i>et al.</i> (2020)	65.65	65.40	59.40
DSP-FWA Li and Lyu (2018)	64.60	57.70	47.20
F ³ -Net Qian <i>et al.</i> (2020)	65.17	-	-
Add-Net Zi <i>et al.</i> (2020)	65.29	-	-
MultiAtt Zhao <i>et al.</i> (2021)	67.02	-	-
RECCE Cao <i>et al.</i> (2022)	68.71	-	-
Proposed + XceptionNet-c0	73.76	72.38	63.08
Proposed + XceptionNet-c23	74.45	73.26	63.00
Proposed + XceptionNet-c40	69.46	69.39	62.65

rithm on the Celeb-DF dataset under the two scenarios- 1) *inter-compression* and 2) *cross-dataset + inter-compression*. Similar to other datasets, high performance is observed on the Celeb-DF dataset for the inter-compression experiment. Table 4.8 shows the results of the inter-compression experiment on the Celeb-DF dataset. For c-0 compression factor, the overall classification accuracy is 86.60% and 89.96% with XceptionNet and ResNet50 as the respective base models. The high performance on the Celeb-DF dataset indicates that the proposed algorithm performs well across different datasets and deep models.

In the cross-dataset+inter-compression experiment, we evaluate the performance of the algorithm on the Celeb-DF dataset with the models trained on the FaceForensics++ dataset. These models are evaluated on c-0, c-23, and c-40 compression factors on the

Celeb-DF dataset and the results are compared with existing algorithms. Table 4.9 compares the AUC scores (%) for the cross-dataset + inter-compression experiment. The results show that the proposed algorithm outperforms the existing algorithms for all three compression factors and achieves state-of-the-art results on the Celeb-DF dataset. Since the model is learning unseen imperceptible artifacts from the FaceForensics++ dataset for increasing the robustness against compression artifacts, it implicitly increases the probability of better generalization over the out-of-distribution unseen Celeb-DF test set. This shows the applicability and generalizability of the proposed algorithm across different datasets for different compression factors.

Most of the existing algorithms work very well on the high quality videos but suffer with high compression ratios. For instance, the proposed algorithm and motion magnification approach (Mehra *et al.*, 2023) yield 72.38% and 72.34% AUC, respectively when evaluated on Celeb-DF and trained on FaceForensics++ with c23 compression. When the compression is increased to c40, the proposed algorithm outperforms the motion magnification approach by over 20%. This is because the compression artifacts are also amplified by the motion magnification step which reduces the gap between real and fake classes. Further, Figure 4.9 shows the results of proposed algorithm on Deepfake samples available in the wild¹. These videos are taken from different social media platforms which employ different levels of compression. The proposed algorithm successfully identifies these videos to be fake. This shows the applicability of the proposed algorithm in real-world scenarios.

The proposed algorithm assumes that all the unseen artifacts are equally important and contribute to improving the model’s performance. However, this assumption may not always be valid, and it requires us to compute the importance of each artifact when compared to seen artifacts. Further, the proposed algorithm is not robust to natural covariates such as pose and tilt. We plan to address these limitations to improve the proposed algorithm in future work.

¹<https://tinyurl.com/2hk3shay>, <https://tinyurl.com/3ptz6smc>, <https://tinyurl.com/3r5ewha3>, <https://tinyurl.com/567wkbh3>, <https://tinyurl.com/4t2ra2eu>,
<https://tinyurl.com/7272pvpn>, <https://tinyurl.com/4zdn9am>, <https://tinyurl.com/47bc94t9>, <https://tinyurl.com/3zrhvzu>, <https://tinyurl.com/2hxhbbkb>,
<https://tinyurl.com/28m4xxze>, <https://tinyurl.com/3enacy3h>, <https://tinyurl.com/rtdmk73m>, <https://tinyurl.com/ywjr8yys>

4.6 Model Robustness to Image Degradations

Although most deepfake detection methods are sensitive to variations in input quality, the proposed algorithm is expected to remain robust under common image degradations such as resizing, blurring, or downsampling. This is because the model does not explicitly rely on compression-specific or pixel-level artifacts but instead captures higher-level inconsistencies that are intrinsic to synthetic content generation. Since these inconsistencies persist across different spatial resolutions and degradation levels, the model’s decision boundary should remain largely unaffected. Nevertheless, a systematic evaluation under diverse image degradation settings would provide a more comprehensive understanding of its generalization ability and confirm the degradation-independence of the proposed approach.

4.7 Summary

With advancements in deep learning approaches, synthesizing fake or manipulated images and videos have become an easy task. These fake or manipulated media can be misused in various ways and therefore, it is important to build classification techniques that can differentiate real content from fake image/video content. In this research, we proposed an algorithm that detects fake content by learning hidden artifacts using the proposed “visibility matrix”. The model built using the visibility matrix is robust to loss of information due to compression. The results demonstrate that the proposed algorithm performs well on compressed images and videos while being trained on high-quality data, thereby being useful for detection at any compression level, once trained. The proposed algorithm can also be useful for various other problems like Deepfake source identification as well as Deepfake duplicacy detection and these problems become more challenging when tested on compressed data. The aforementioned research directions require the attention of the research community to prevent AI from being misused and build trusted AI systems.

Chapter 5

Data Fine-tuning

5.1 Introduction

With the advancements in machine learning (specifically deep learning), ready to use Commercial Off-The-Shelf (COTS) systems are available for automated face analysis, such as face recognition (Ding and Tao, 2018), emotion recognition (Fan *et al.*, 2016), and attribute prediction (Hand *et al.*, 2018). However, often times the details of the model are not released which makes it difficult to update it for any other task or datasets. This renders the model's effectiveness as a black-box model only. To illustrate this, let $\mathbf{X}$ be the input data for a model with weights $\mathbf{W}$ and bias b . This model can be expressed as:

$$\phi(\mathbf{W}\mathbf{X} + b) \tag{5.1}$$

If the source of the model is available, model fine-tuning is used to update the parameters. However, as mentioned above, in black box scenarios, the model parameters, $\mathbf{W}$ and b cannot be modified, as the user does not have access to the model.

“Can we enhance the performance of a black-box system for a given dataset?” To answer this question, in this research, we present a novel concept termed as **Data Fine-tuning (DFT)**, wherein the input data is adjusted corresponding to the model's unseen decision boundary. To the best of our knowledge, this is the first work towards data fine-tuning to enhance the performance of a given black box system. As shown in Figure 5.1, the proposed data fine-tuning adjusts the input data $\mathbf{X}$ whereas, in the model fine-tuning approach (MFT), the parameters ($\mathbf{W}$, b) are adjusted for optimal classification.

Mathematically, model fine-tuning is:

$$\phi(\mathbf{W}\mathbf{X} + b) \xrightarrow{\text{MFT}} \phi(\mathbf{W}'\mathbf{X} + b') \tag{5.2}$$

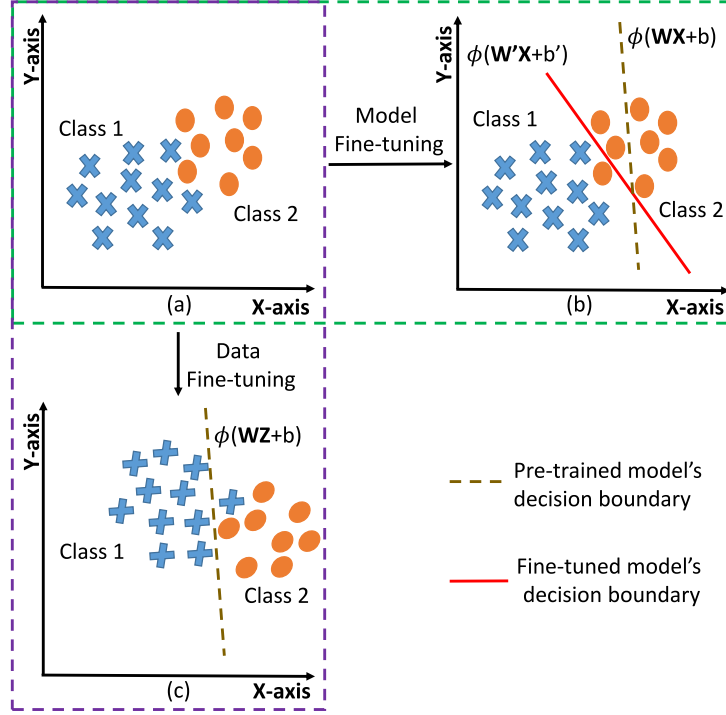


Figure 5.1: Illustration of model fine-tuning and data fine-tuning: (a) represents the data distribution with two classes. (b) represents Model Fine-tuning where the model's decision boundary shifts corresponding to the input data, and (c) represents Data Fine-tuning where the input data shifts corresponding to model's decision boundary.

and data fine-tuning can be written as:

$$\phi(\mathbf{W}\mathbf{X} + b) \xrightarrow{\text{DFT}} \phi(\mathbf{W}\mathbf{Z} + b) \quad (5.3)$$

where, MFT and DFT are model fine-tuning¹ and data fine-tuning, respectively. $(\mathbf{W}', b')$ are the parameters after MFT and $\mathbf{Z}$ is the perturbed version of input $\mathbf{X}$ after data fine-tuning.

In this research, the proposed data fine-tuning is achieved using adversarial perturbation. For this purpose, samples in the training data are uniformly perturbed and the model is trained iteratively on this perturbed training data to minimize classification loss. After each iteration, optimization is performed over the perturbation noise and added to the training data. At the end of the training, a single uniform perturbation is learned corresponding to a dataset. As a case study, the proposed algorithm is evaluated for facial attribute classification. It learns a single universal perturbation for a given

¹Various data augmentation techniques have also been used for model fine-tuning (Salamon and Bello, 2017; Um *et al.*, 2017; Wu *et al.*, 2018a)

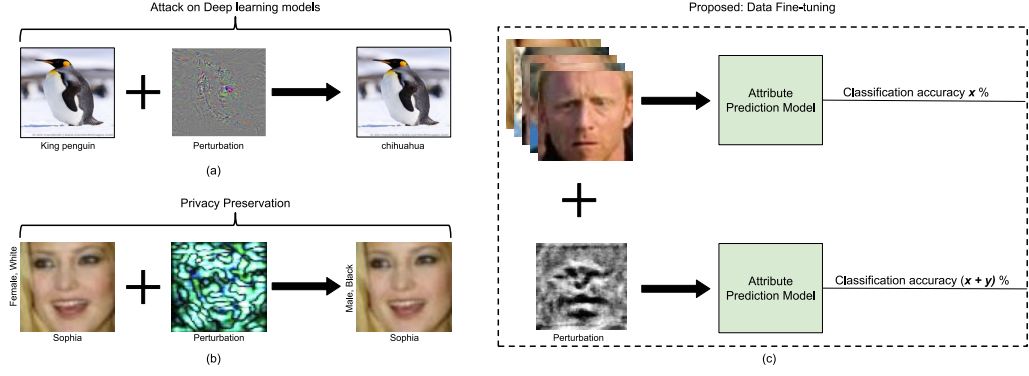


Figure 5.2: Comparing the concept of adversarial perturbation with data fine-tuning. (a) Adversarial perturbation: shows the application of perturbation in attacking deep learning models (Xie *et al.*, 2017). (b) Privacy preservation: perturbation can be used to anonymize the attributes by preserving the identity of the input image (Chhabra *et al.*, 2018). (c) Data Fine-tuning: illustrates the proposed application of perturbation in enhancing the performance of a model.

dataset to improve facial attribute classification while preserving the visual appearance of the images. Experiments are performed on three publicly available datasets and results showcase enhanced performance of black box systems using data fine-tuning.

5.1.1 Related Work

In the literature, perturbation is studied from two perspectives: (i) privacy preservation and (ii) attacks on deep learning models. For privacy preservation, several techniques utilizing data perturbation are proposed. Jain and Bhandare (2011) proposed min max normalization method to perturb data before using in data mining applications. Last *et al.* (2014) proposed a data publishing method using NSVDist. Using this method, the sensitive attributes of the data are published as the frequency distributions. Recently, Chhabra *et al.* (2018) proposed an algorithm to anonymize multiple facial attributes in an input image while preserving the identity using adversarial perturbation. Li and Zhou (2018) proposed Random Linear Transformation with Condensed Information-Support Vector Machine to convert the condensed information to another random vector space to achieve safe and efficient data classification.

Szegedy *et al.* (2013) demonstrated that application of imperceptible perturbation could lead to the misclassification of an image. Papernot *et al.* (2016b) created an adversarial attack by restricting l_0 -norm of the perturbation where only a few pixels of an

image are modified to fool the classifier. Carlini and Wagner (2017) introduced three adversarial attacks and showed the failure of defensive distillation Carlini and Wagner (2016) for targeted networks. By adding perturbation, Kurakin *et al.* (2016) replaced the original label of the image with the label of least likely predicted class by the classifier. This lead to the poor classification accuracy of Inception v3. Su *et al.* (2017) proposed a one-pixel attack in which three networks are fooled by changing one pixel per image. Universal adversarial perturbation proposed by Moosavi-Dezfooli *et al.* (2017) can fool a network when applied to any image. This overcomes the limitation of computing perturbation on every image. Goswami *et al.* (2018) proposed a technique for automatic detection of adversarial attacks by using the abnormal filter response from the hidden layer of the deep neural network. Further, a novel technique of selective dropout is proposed to mitigate the adversarial attacks. Goel *et al.* (2018) developed SmartBox toolbox for detection and mitigation of adversarial attacks against face recognition. Wei *et al.* (2023) introduce a single adversarial patch designed to simultaneously deceive both visible and infrared object detectors by optimizing the patch’s shape and leveraging score-aware iterative evaluation. Fan *et al.* (2024) proposed a method leveraging locally estimated global perturbations to enhance adversarial effectiveness in federated learning scenarios. Gao *et al.* (2024) introduce a label-free white-box attack targeting Vision Transformer (ViT) models by exploiting the feature collapse phenomenon, where attention mechanisms overly depend on low-frequency components, leading to similar features in deeper layers.

Existing literature demonstrates the application of adversarial perturbation for performing attacks on deep learning models and in privacy preservation (Figure 5.2(a) and (b)). However, data fine-tuning using adversarial perturbation (Figure 5.2(c)) for enhancing the performance of a model is not yet explored.

5.2 Proposed Approach: Data Fine-tuning

Considering a black-box system as a pre-trained model, the problem statement can be defined as “given the dataset $\mathbf{D}$ and pre-trained model M , learn a perturbation vector $\mathbf{N}$ such that adding noise $\mathbf{N}$ to $\mathbf{D}$ improves the performance of the model M on $\mathbf{D}$ ”. There are two important considerations while performing data fine-tuning:

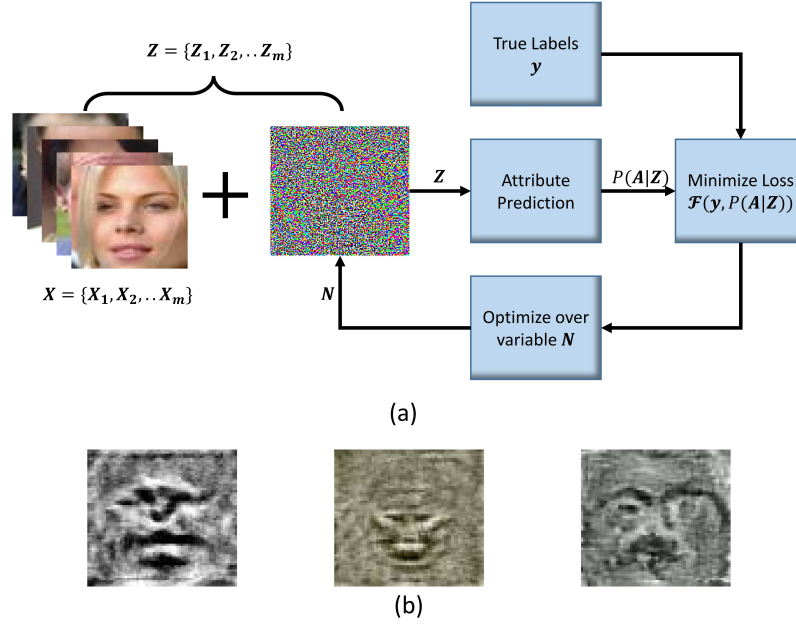


Figure 5.3: (a) Block diagram illustrating the steps of the proposed algorithm. In the first step, perturbation is initialized with zero image and added to the original training data. In the next step, perturbed training data is given as input to the (attribute prediction) model followed by the computation of loss. After that, optimization is performed over perturbation and added to the training data. (b) Some samples of the learned perturbation using the proposed algorithm. The first two visualizations correspond to the perturbation learned for ‘Smiling’ attribute of LFW and CelebA datasets, respectively. The third visualization corresponds to the ‘Gender’ attribute of the MUCT dataset.

1. To learn a single universal perturbation noise for a given dataset.
2. The visual appearance of the image should be preserved after performing data fine-tuning.

The block diagram illustrating the steps involved in the proposed algorithm is shown in Figure 5.3. The optimization process for data fine-tuning using adversarial perturbation with applications to facial attribute classification is discussed below. This same approach can be extended for other classification models.

Given the original training set $\mathbf{X}$ with m number of images where each image, $\mathbf{X}_k$ has pixel values in the range $\{0, 1\}$, i.e., $\mathbf{X}_k \in [0, 1]$. Let $\mathbf{Z}$ be the perturbed training set generated by adding model specific perturbation noise $\mathbf{N}$ such that the pixel values of each output perturbed image $\mathbf{Z}_k$ ranges between 0 to 1, i.e., $\mathbf{Z}_k \in [0, 1]$. Mathematically, it is written as:

$$\mathbf{Z}_k = f(\mathbf{X}_k + \mathbf{N}) \quad (5.4)$$

such that $f(\mathbf{X}_k + \mathbf{N}) \in [0, 1]$

where, $f(\cdot)$ represents the function to transform an image in the range of 0 to 1. In order to satisfy the above constraint, inspired by Carlini and Wagner (2017), the following function is used:

$$\mathbf{Z}_k = \frac{1}{2}(\tanh(\mathbf{X}_k + \mathbf{N}) + 1) \quad (5.5)$$

For each image $\mathbf{X}_k$ there are n number of attributes in the attribute set $\mathbf{A}$, where each attribute $\mathbf{A}_i$ has C_j number of classes. For example, ‘Gender’ attribute has two classes namely {Male, Female} while ‘Expression’ attribute has three classes namely {Happy, Sad, Anger}. Mathematically, it is written as:

$$\mathbf{A} = \{\mathbf{A}_1(C_1), \mathbf{A}_2(C_2), \dots, \mathbf{A}_n(C_n)\} \quad (5.6)$$

The pre-trained attribute prediction model for attribute $\mathbf{A}_i$ is represented as $\phi_{A_i}(\mathbf{X}_k, \mathbf{W}, b)$, where $\mathbf{W}$ is the weight matrix, and b is the bias. The output attribute score of any image $\mathbf{X}_k$ is written as:

$$P(\mathbf{A}_i|\mathbf{X}_k) = \phi_{A_i}(\mathbf{X}_k, \mathbf{W}, b) \quad (5.7)$$

where, $P(\mathbf{A}_i|\mathbf{X}_k)$ represents the output attribute score of the input image $\mathbf{X}_k$ for attribute $\mathbf{A}_i$. In order to perform data fine-tuning, perturbation $\mathbf{N}$ is added to each input image $\mathbf{X}_k$ to get the output perturbed image $\mathbf{Z}_k$ using Equation 5.5. Here, $\mathbf{N}$ is the perturbation variable to be optimized. The output attribute score of the perturbed image $\mathbf{Z}_k$ is represented as:

$$P(\mathbf{A}_i|\mathbf{Z}_k) = \phi_{A_i}(\mathbf{Z}_k, \mathbf{W}, b) \quad (5.8)$$

In order to enhance the model’s performance for attribute $\mathbf{A}_i$, the distance between the true class and attribute predicted score of the perturbed image is minimized which is expressed as:

$$\min_{\mathbf{N}} \mathcal{F}(\mathbf{y}_{i,k}, P(\mathbf{A}_i|\mathbf{Z}_k)) \quad (5.9)$$

where, $\mathcal{F}(\cdot, \cdot)$ represents the function to minimize the distance between the true class and the predicted class. $\mathbf{y}_{i,k}$ represents the true class of attribute $\mathbf{A}_i$ in one hot encoding form of the original image $\mathbf{X}_k$. To preserve the visual appearance of the output perturbed image $\mathbf{Z}_k$, the distance between original image $\mathbf{X}_k$ and the perturbed image $\mathbf{Z}_k$ is minimized. Thus, the above equation is updated as:

$$\min_{\mathbf{N}} \mathcal{F}(\mathbf{y}_{i,k}, P(\mathbf{A}_i|\mathbf{Z}_k)) + H(\mathbf{X}_k, \mathbf{Z}_k) \quad (5.10)$$

where, H represents the distance metric to minimize the distance between $\mathbf{X}_k$ and $\mathbf{Z}_k$. In this research, Euclidean distance metric is used to preserve the visual appearance of the image. Therefore,

$$\min_{\mathbf{N}} \mathcal{F}(\mathbf{y}_{i,k}, P(\mathbf{A}_i|\mathbf{Z}_k)) + \|\mathbf{X}_k - \mathbf{Z}_k\|_F^2 \quad (5.11)$$

Since the output class score ranges between 0 and 1, the objective function in Equation (5.9) is formulated as:

$$\mathcal{F}(\mathbf{y}_i, P(\mathbf{A}_i|\mathbf{Z})) = \frac{1}{m} \sum_{k=1}^m \max(0, 1 - \mathbf{y}_{i,k}^T P(\mathbf{A}_i|\mathbf{Z}_k)) \quad (5.12)$$

where, $i \in \{1, \dots, n\}$, and the term $\mathbf{y}_{i,k}^T P(\mathbf{A}_i|\mathbf{Z}_k)$ outputs the attribute score of the true class. As the above function $\mathcal{F}(\mathbf{y}_i, P(\mathbf{A}_i|\mathbf{Z}))$ is to be minimized, the term $\max(0, 1 - \mathbf{y}_{i,k}^T P(\mathbf{A}_i|\mathbf{Z}_k))$ enforces the output attribute score of the true class of the perturbed image $\mathbf{Z}_k$ towards one.

Figure 5.4 illustrates the proposed algorithm with an example. Let $\mathbf{D}_1$ be the dataset with two classes in the input image space (Figure 5.4(a)) and it is used to train a model, M_1 . Model M_1 computes the decision boundary and projects the output class scores corresponding to the input data $\mathbf{D}_1$ as shown in Figure 5.4(b). It is observed that the output class scores are well separated across the decision boundary for the dataset $\mathbf{D}_1$. Now, the pre-trained model M_1 is used for projecting the input dataset $\mathbf{X}$ (Figure 5.4(c)). The decision boundary of the model M_1 remains fixed. The projected output class scores of the input data $\mathbf{X}$ are shown in Figure 5.4(d). It is observed that most of the data points of both the classes are projected on the same side of the decision boundary resulting in a high classification error. This is due to the change in the data distribution of the

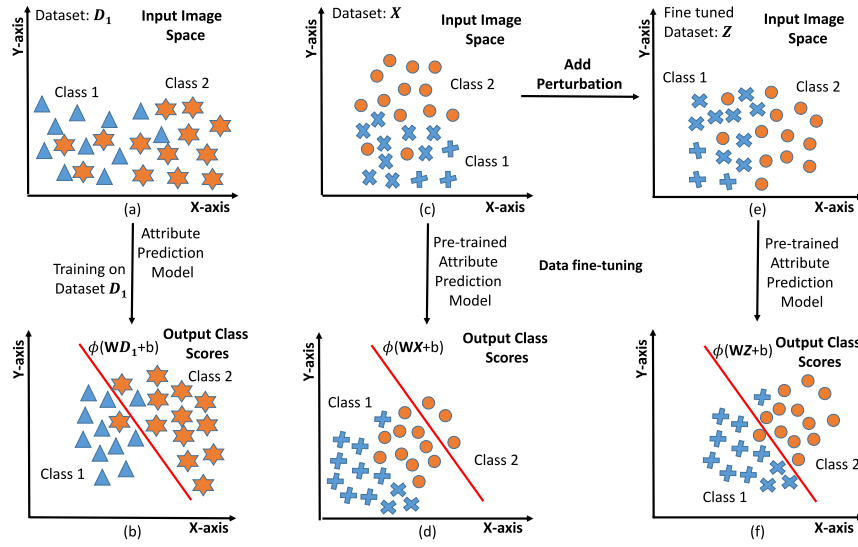


Figure 5.4: Illustration of the proposed DFT algorithm. Figure (a)-(b) represents the training of attribute prediction model using dataset D_1 . (c)-(d) shows the performance of the trained attribute prediction model on dataset X . (e)-(f) shows the performance of the fine-tuned dataset Z by adding perturbation on trained attribute prediction model.

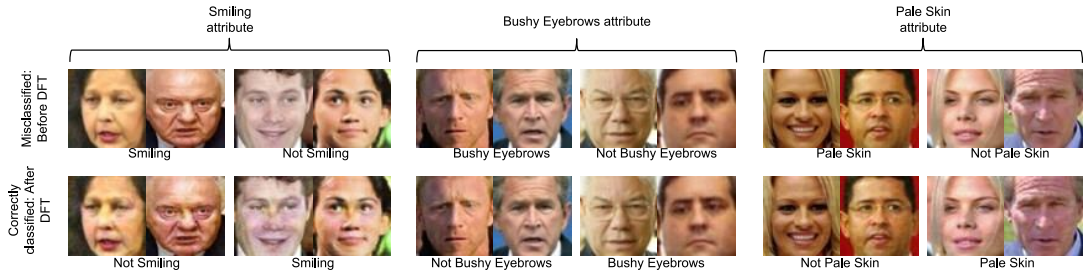


Figure 5.5: Misclassified samples that are correctly classified after data fine-tuning. First row shows the images misclassified before data fine-tuning while the second row represents their correct classification after data fine-tuning. The first block of images correspond to the ‘Smiling’ attribute, second block corresponds to ‘Bushy Eyebrows’, while the third block corresponds to ‘Pale Skin’ of the LFW dataset.

input dataset X . To overcome this problem, input dataset X is fine-tuned by adding perturbation noise. Figure 5.4(e) shows the fine-tuned dataset Z that is given as input to the model M_1 . The projection of the fine-tuned dataset Z is shown in Figure 5.4(f). On comparing the output class scores of the projection of input data X and fine-tuned data Z , it is observed that several misclassified samples from X are correctly classified with the fine-tuned dataset Z .

Table 5.1: Details of the experiments to show the efficacy of the proposed data fine-tuning for facial attribute classification.

Experiment		Data Fine-tuning	Model Training
	Attribute	Database	Database
Black Box Data Fine-tuning: Intra Dataset	Gender	MUCT	MUCT
		LFW	LFW
		CelebA	CelebA
	Smiling, Bushy Eyebrows Pale Skin	LFW	LFW
	Smiling, Attractive, Wearing Lipstick	CelebA	CelebA
Black Box Data Fine-tuning: Inter Dataset	Gender	MUCT	LFW, CelebA
		LFW	MUCT, CelebA
		CelebA	MUCT, LFW
	Smiling, Bushy Eyebrows, Pale Skin	LFW	CelebA
	Smiling, Attractive, Wearing Lipstick	CelebA	LFW

5.3 Datasets Protocol and Experimental Details

The proposed algorithm is evaluated on three publicly available datasets for facial attribute classification: LFW Huang *et al.* (2008), CelebA Liu *et al.* (2015a), and MUCT Milborrow *et al.* (2010b). A comparison has also been performed between Data Fine-tuning and Model Fine-tuning. The details of each dataset and its protocol are described below :

LFW dataset consists of 13,133 images of 5,749 subjects. Total 73 attributes are annotated with intensity values for each image. The attributes are binarized by considering positive intensity values as attribute present with label 1 and negative intensity values as attribute absent with label 0. The dataset is partitioned into 60% training set, 20% validation set, and 20% testing set.

CelebA dataset consists of 202,599 face images of more than 10,000 celebrities. For each image, 40 binary attributes are annotated such as Male, Smiling, and Bushy Eyebrows. Standard pre-defined protocol is followed for experiments and the dataset

Table 5.2: Classification accuracy (in %) of before and after Data Fine-tuning(DFT) for ‘Gender’ attribute on LFW, CelebA, and MUCT datasets.

	Before DFT	After DFT
LFW	87.94	91.17
CelebA	82.13	83.08
MUCT	91.67	94.31

Table 5.3: Classification accuracy (in %) before and after performing data fine-tuning for three attributes on the LFW and CelebA datasets.

LFW	Smiling		Bushy Eyebrows		Pale Skin	
	Before	After	Before	After	Before	After
	76.18	82.42	68.34	69.98	72.83	74.81
CelebA	Smiling		Attractive		Wearing Lipstick	
	Before	After	Before	After	Before	After
	67.82	71.30	70.48	70.54	80.95	81.29

is partitioned into 162,770 images in the training set, 19,867 into validation set, and 19,962 images in the testing set.

MUCT dataset consists of 3,755 images of 276 subjects out of which 131 are male and 146 are female. Viola-Jones face detector is applied on all the images, and the detector failed to detect 49 face images. Therefore, only 3,706 images are considered for further processing. These images are further partitioned into 60% training set, 20% validation set, and 20% testing set corresponding to each class.

To evaluate the performance of data fine-tuning, two experiments are performed, (i) *Black Box Data Fine-tuning: Intra Dataset* and (ii) *Black Box Data Fine-tuning: Inter Dataset*. Both the experiments are performed on all the three datasets. Classification performance of the attributes is enhanced corresponding to the attribute classification model. To train the attribute classification model, pre-trained VGGFace Parkhi *et al.* (2015) + NNET is used. Experimental details are also shown in Table 5.1.

5.3.1 Implementation Details

The implementation details of training attribute classification model, perturbation learning, and model fine-tuning are discussed below.

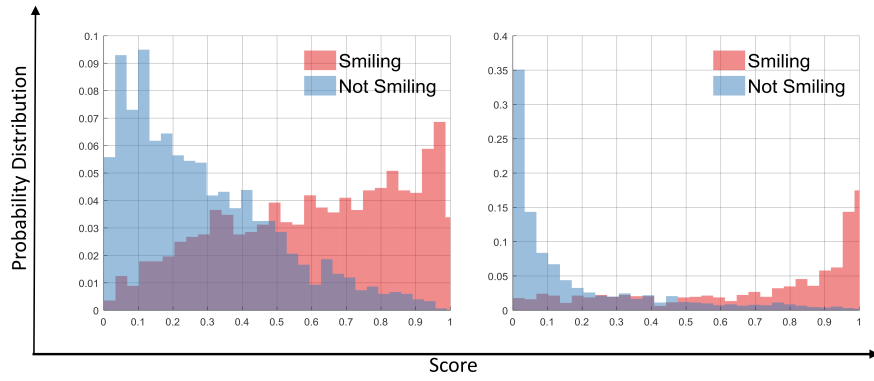


Figure 5.6: Smiling attribute score distribution pertaining to before and after performing data fine-tuning on the LFW dataset. The left graph represents the score distribution before data fine-tuning and right graph represents the score distribution after data fine-tuning.

Table 5.4: Confusion matrix of the LFW dataset for three attributes: ‘Smiling’, ‘Bushy Eyebrows’, ‘Pale Skin’.

			Attribute Class	Prediction		Attribute Class	Prediction		Attribute Class	Prediction	
				Smiling	Not Smiling		Bushy Eyebrows	Not Bushy Eyebrows		Pale Skin	Not Pale Skin
LFW	Ground Truth	Before Data Fine-tuning	Smiling	65.50	34.50	Bushy Eyebrows	77.17	22.83	Pale Skin	74.48	25.52
			Not Smiling	15.86	84.14	Not Bushy Eyebrows	42.23	57.77	Not Pale Skin	28.75	71.25
		After Data Fine-tuning	Smiling	73.26	26.74	Bushy Eyebrows	79.19	20.81	Pale Skin	76.57	23.43
			Not Smiling	10.76	89.24	Not Bushy Eyebrows	41.05	58.95	Not Pale Skin	26.88	73.12

Training Attribute Classification Model: To train attribute classification model pre-trained VGGFace+NNET is used. Two fully connected layers are used for training NNET of 512 dimensions. Each model is trained for 20 epochs with Adam optimizer, and learning rate is set to 0.005.

Perturbation Learning: To learn the perturbation for a given dataset, learning rate is set to 0.001 and the batch size is 800. The number of iterations used for processing each batch is 16, and the number of epochs is 5.

Model Fine-tuning: To fine-tune the attribute classification model, Adam optimizer is used with learning rate set to 0.005. The model is trained for 20 epochs.

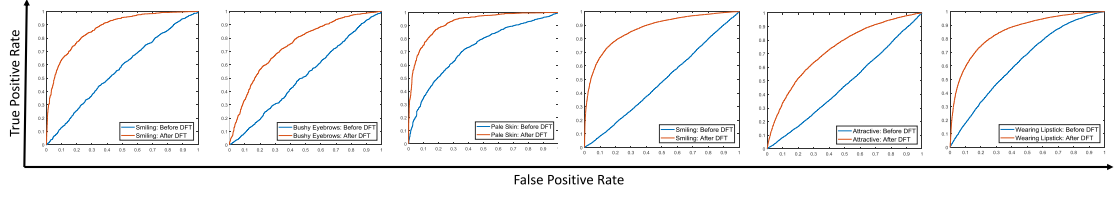


Figure 5.7: ROC plots showing before and after data fine-tuning results of Black box Data Fine-tuning: Inter Dataset Experiment. First three ROC curves shows the result on the LFW dataset using a model trained on the CelebA dataset. Last three ROC curves shows the result on the CelebA dataset using a model trained on the LFW dataset.

5.4 Performance Evaluation

The performance of the proposed algorithm is evaluated for *Black Box Data Fine-tuning: Intra Dataset Experiment*, where the dataset used for data fine-tuning is same on which the pre-trained model is trained. On the other hand, in *Black Box Data Fine-tuning: Inter Dataset Experiment*, the training data used to perform data fine-tuning is different from the training data used to train the pre-trained model.

5.4.1 Black Box Data Fine-tuning: Intra Dataset Experiment

The proposed algorithm is evaluated on LFW, CelebA, and MUCT datasets for enhancing the performance of black box models. ‘Gender’ is the common attribute among all three datasets. Table 5.2 shows the classification accuracy pertaining to before and after data fine-tuning for ‘Gender’ attribute. For all three datasets, the classification accuracy improves by 1% to 3% using data fine-tuning. Specifically, the classification accuracy increases by 2.64% for MUCT dataset whereas, for LFW dataset, the accuracy increases by 3.21%.

Three additional attributes, namely LFW-{'Smiling', 'Bushy Eyebrows', 'Pale Skin'}, CelebA-{'Smiling', 'Attractive', 'Wearing Lipstick'} are also evaluated. Table 5.3 shows the classification accuracy corresponding to these attributes. Similar to the results on ‘Gender’ attribute, data fine-tuning leads to an overall increase in the classification accuracies of all the attributes for both the datasets. The classification accuracy of ‘Smiling’ attribute increases by approximately 6% for LFW dataset and 4% for CelebA dataset. This shows the utility of data fine-tuning in enhancing the model’s performance trained on the same dataset.

Table 5.5: Classification accuracy(%) of Black box Data Fine-tuning: Inter Dataset experiment for ‘Gender’ attribute on the MUCT, LFW, and CelebA datasets.

		Dataset used to train the model					
		MUCT		LFW		CelebA	
		Before	After	Before	After	Before	After
Dataset	MUCT	-	-	57.84	83.65	80.27	92.84
	LFW	63.09	80.45	-	-	56.01	86.33
	CelebA	49.14	74.73	67.53	76.59	-	-

Table 5.6: Classification accuracy(%) of Black box Data Fine-tuning: Inter Dataset experiment.

	Pre-trained Model trained on CelebA					
	Smiling		Bushy Eyebrows		Pale Skin	
	Before	After	Before	After	Before	After
LFW	55.29	78.61	45.40	68.91	56.62	84.21
	Pre-trained Model trained on LFW					
	Smiling		Attractive		Wearing Lipstick	
	Before	After	Before	After	Before	After
CelebA	49.07	66.97	49.71	66.60	60.25	77.15

Figure 5.5 shows some misclassified samples of LFW dataset corresponding to ‘Smiling’, ‘Bushy Eyebrows’, and ‘Pale Skin’ attributes that are correctly classified after data fine-tuning. It is also observed that the visual appearance of the images is preserved. The score distribution of ‘Smiling’ attribute, before and after data fine-tuning is shown in Figure 5.6. It is observed that the overlapping region between both the classes is reduced, and the confidence of predicting the true class scores is increased after data fine-tuning. The confusion matrix corresponding to the three attributes of the LFW dataset is shown in Table 5.4 which indicates that the True Positive Rate (TPR) and True Negative Rate (TNR) is improved for all three attributes. For instance, the TPR of ‘Smiling’ attribute is increased by approximately 8% and TNR is increased by approximately 5% showcasing the efficacy of the proposed technique.

5.4.2 Black box Data Fine-tuning: Inter Dataset Experiment

This experiment is performed considering the real world scenario associated with Commercial off-the-shelf (COTS) systems where the training data distribution of the system

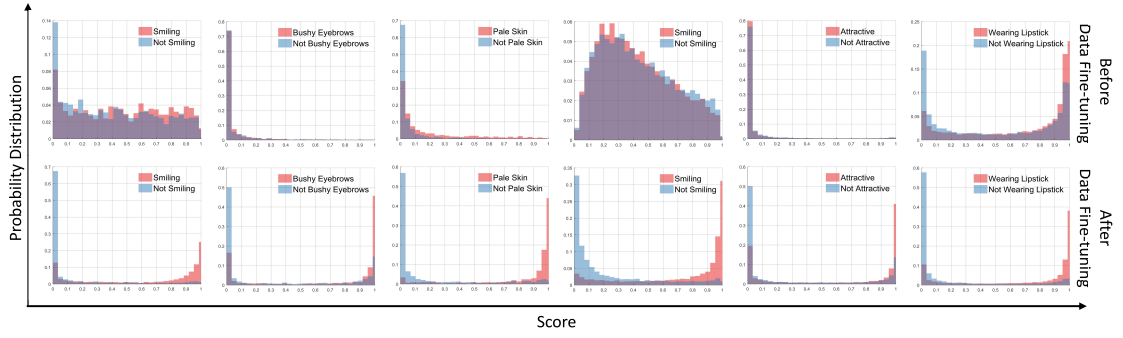


Figure 5.8: Score distributions pertaining to before and after performing data fine-tuning. Top three graphs from the left represent the distribution of the LFW dataset predicted using a model trained on the CelebA dataset. Bottom three graphs from the left represent its corresponding distribution after data fine-tuning. Similarly top three graphs from the right represent the score distribution on the CelebA dataset predicted using a model trained on the LFW dataset. Bottom three graphs from the right represent its corresponding distribution after data fine-tuning.

is unknown to the user. The performance is evaluated for ‘Gender’ attribute on all three datasets and the other three attributes used in Experiment 1 for LFW and CelebA datasets.

Table 5.5 shows the classification accuracies for ‘Gender’ attribute. It is observed that the classification accuracies increase by 12% to 30% on all three datasets. For other attributes on LFW and CelebA datasets, data fine-tuning is performed on the LFW dataset using a model trained on the attributes of the CelebA dataset and vice versa. Classification accuracies in Table 5.6 show the significant enhancement in the performance of the black box system using data fine-tuning. For instance, the accuracy on ‘Bushy Eyebrows’ of the LFW dataset increases by approximately 23%. Similarly, there is an improvement of 17% on the attribute ‘Attractive’ of the CelebA dataset. Figure 5.7 shows the ROC plots of all three attributes of LFW and CelebA datasets. The significant difference in the curves for all the attributes clearly demonstrates that the proposed algorithm is capable of improving the performance of the model with a large margin. Figure 5.8 shows the score distribution before and after applying data fine-tuning. It is observed that before data fine-tuning, there is a huge overlap among the distributions of both the classes. For instance, the distribution of the attribute ‘Bushy Eyebrows’ before perturbation for both the classes is on the same side resulting in higher misclassification rate. After data fine-tuning, the distribution of both the classes is well separated. This illustrates that data fine-tuning is able to shift the data corresponding to

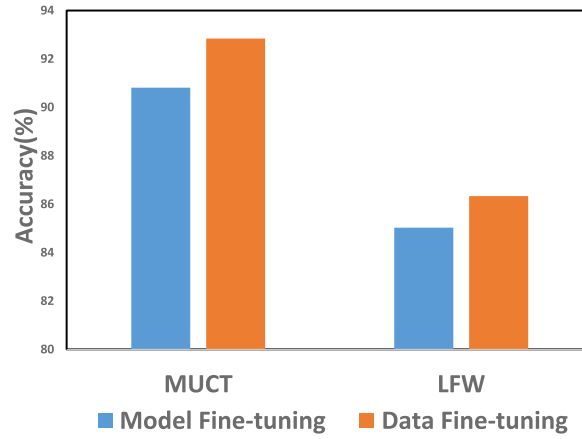


Figure 5.9: Comparing Data Fine-tuning versus Model Fine-tuning for ‘Gender’ attribute on the MUCT and LFW datasets using a model trained on the CelebA dataset.

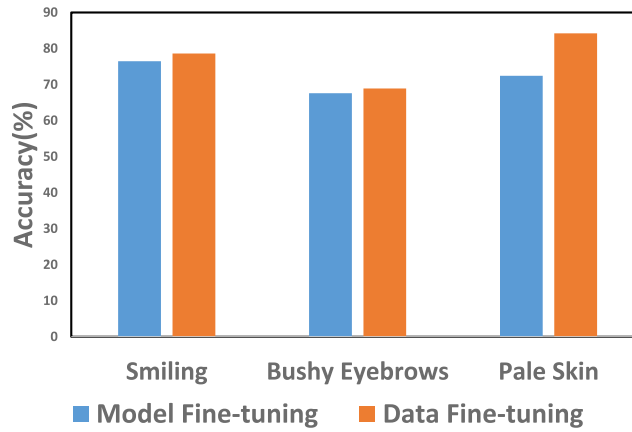


Figure 5.10: Comparing the results of Data Fine-tuning versus Model Fine-tuning on the LFW dataset using a model trained on the CelebA dataset.

the model’s unseen decision boundary.

5.4.3 Model Fine-tuning versus Data Fine-tuning

This experiment is performed to compare the performance of model fine-tuning, where the model acts as a white box versus data fine-tuning where the model is a black box. For the experiments related to data fine-tuning, the procedure of ‘Black Box Data Fine-tuning: Inter Dataset Experiment’ is followed. For model fine-tuning, the attribute classification model trained on the CelebA dataset is fine-tuned with MUCT and LFW dataset. Figure 5.9 shows the comparison of data fine-tuning with model fine-tuning for ‘Gender’ attribute. In this experiment, the pre-trained model is trained on the CelebA dataset. On comparing the results on MUCT and LFW datasets, it is observed that data fine-tuning performs better than model fine-tuning for both the datasets. Experimental

results obtained with other three attributes are shown in Figure 5.10, which also indicate that data fine-tuning outperforms model fine-tuning. Experiments are also performed by combining model fine-tuning with data fine-tuning. For this purpose, an iterative approach is followed, where data fine-tuning and model fine-tuning are performed iteratively. It is observed that the combination of model fine-tuning and data fine-tuning further enhances the results. However, such a combination is not useful for black-box systems where model fine-tuning is not possible.

5.4.4 Robustness against New Biases and Distortions

DFT is designed not only to improve model performance but also to safeguard against bias or distortion. In DFT, a single universal perturbation is learned per dataset to uniformly nudge all samples toward better alignment with the model’s decision boundary. The perturbation is constrained to be imperceptible, preserving the visual fidelity of attributes (e.g., a “Bushy Eyebrows” sample remains visually bushy, while only classifier confidence improves). By avoiding per-sample optimization, DFT minimizes the risk of distorting minority subgroups disproportionately, and experiments show that class score distributions become more separable without altering attribute semantics.

5.5 Model Transferability of Learned Perturbation

The proposed data fine-tuning framework learns a universal perturbation tailored for a specific model retrained on a given dataset to enhance its performance on a target dataset. However, the learned perturbation is inherently model-dependent and may not generalize well to other architectures or classifiers. Since different models often exhibit distinct feature representations and decision boundaries, the perturbation optimized for one model may not yield similar improvements when applied to another. Investigating the cross-model transferability of the learned perturbation and developing model-agnostic or ensemble-based fine-tuning strategies represent promising directions for future research.

5.6 Summary

Increasing demands of automated systems for face analysis has led to the development of several COTS systems. However, COTS systems are generally provided as black box systems and the model parameters are not available. In such scenarios, enhancing the performance of black-box systems is a challenging task. To address this situation, in this research a novel concept of data fine-tuning is proposed. Data fine-tuning refers to the process of adjusting the input data according to the behavior of the pre-trained model. The proposed data fine-tuning algorithm is designed using adversarial perturbation. Multiple experiments are performed to evaluate the performance of the proposed algorithm. It is observed that data fine-tuning enhances the performance of black box models. A comparison of data fine-tuning with model fine-tuning is also performed. We postulate that data fine-tuning can be an exciting alternative to model fine-tuning, particularly for black-box systems.

Chapter 6

Feature Guided Perturbation for Facial Attribute Classification

6.1 Introduction

With increasing applications of machine learning and deep learning, addressing data-shift is of paramount importance. Pre-trained models generally do not yield good performance if the target dataset is different from the pre-training dataset. In order to address the problem of data shift, several techniques such as metric learning and transfer learning are used to optimize the feature space by updating the model on the target dataset. However, updating deep models require millions of parameters update and are computationally expensive.

In our previous research, the concept of data fine-tuning has been presented, where the input data is adjusted corresponding to the model's unseen classification decision boundary (Chhabra *et al.*, 2019). Motivated by this, in this research, we postulate that “*feature space of a pre-trained model can be optimized without updating model parameters*”. To achieve this, we propose an algorithm to learn a single uniform perturbation for a given dataset. The learned perturbation, when added to the input dataset and given as input to the pre-trained model, results in an optimized feature space. In this research, the data obtained after adding the learned perturbation is termed as **Feature Guided Perturbed (FGP)** data. Also, the FGP data obtained using the proposed algorithm acts as a supplement to model fine-tuning to boost the classification performance.

Figure 6.1 illustrates the concept of data shift on a synthetically generated 2D dataset. As shown in Figure 6.1, *Dataset B* in the input space results in overlapped feature space. However, the addition of a learned perturbation (obtained using the proposed algorithm) to *Dataset B* shifts the data in the input space toward the region where the output features are separable. The proposed algorithm learns the perturbation corresponding to a loss function which pulls the samples of the same class towards its class

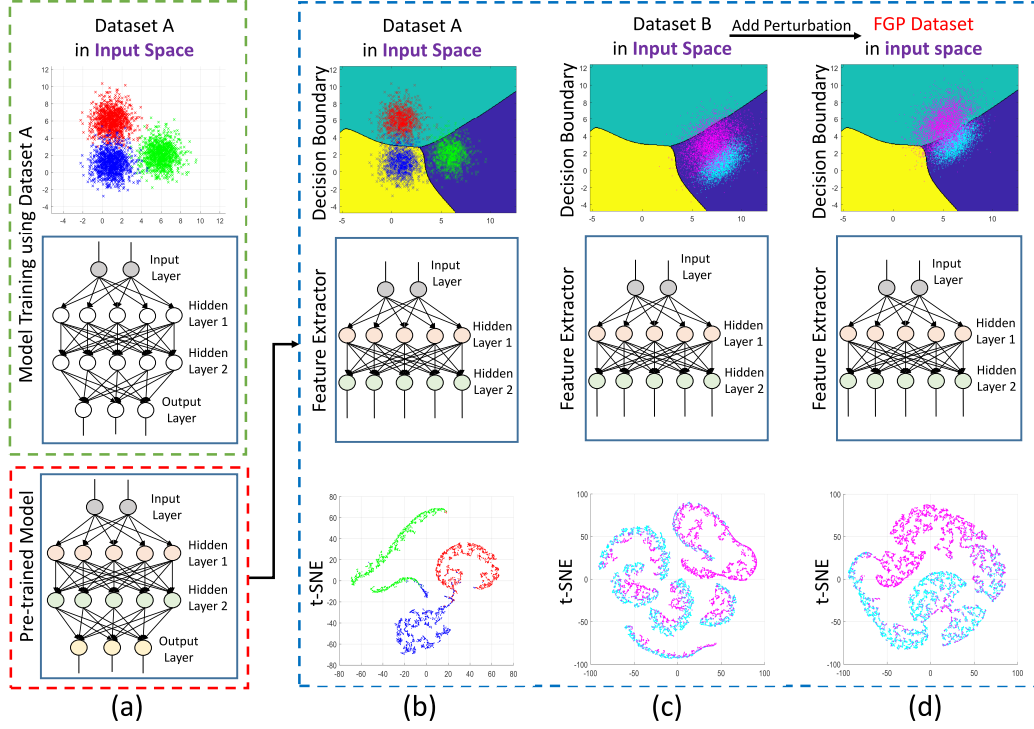


Figure 6.1: Illustrating the concept of data shift in the input space on a synthetically generated 2D dataset. (a) Demonstrates the model training on *Dataset A* with three classes. (b) Shows *Dataset A* with decision boundaries in the input space corresponding to the pre-trained model and t-SNE representation of the features obtained using the pre-trained model. (c) A new two-class *Dataset B* with unseen decision boundaries in the input space along with t-SNE representation corresponding to the pre-trained model (on *Dataset A*). (d) Shows the Feature Guided Perturbed (FGP) dataset obtained after adding the learned perturbation to the *Dataset B* mentioned in (c) along with its t-SNE representation.

center and pushes away the other class centers in the feature space. In addition to this, a method termed as **Feature Drop** is proposed to minimize the effect of outliers while computing the class center. The novelty of the proposed algorithm is that it optimizes the feature space without updating model parameters. The key contributions are:

- A novel algorithm is proposed to shift the input data via perturbation learning (FGP data) for feature space optimization. The proposed algorithm extends the role of adversarial perturbation and shows that perturbations can be used to improve the performance of pre-trained models without fine-tuning them. Also, the proposed algorithm acts as a supplement to model fine-tuning to further boost the overall performance.
- Class center in feature space is generally computed by taking the mean output feature representation of a class. However, the presence of outliers can affect the mean of a particular class. Therefore, to minimize the effect of outliers, we propose “Feature Drop,” in which the output feature representations in the feature space are randomly dropped to compute the center towards the denser region.

- The number of learned parameters for obtaining the FGP data is equal to the size of the input image.
- The performance of the proposed algorithm is evaluated for facial attribute classification. The proposed algorithm not only improves the performance of models for single attribute classification but also shows improvement for multiple attribute classification.

6.2 Literature Review

The proposed research relates to three areas in the literature: (i) feature space optimization, (ii) perturbation learning, and (iii) facial attribute classification. We have summarized the literature in the following subsections.

6.2.1 Feature Space Optimization

For feature space optimization, different techniques have been proposed, such as projection, manifold learning, metric learning, and deep neural networks, to transform the data from input space to a different feature space where the classes are separable. In deep learning research, Schroff *et al.* (2015) proposed FaceNet to learn a mapping from face images to a compact Euclidean space where distances directly correspond to a measure of face similarity. Kundu *et al.* (2016) optimized the mapping of pixels to a Euclidean feature space to minimize the distance between the corresponding points for semantic video segmentation. Shi *et al.* (2016) proposed a class-encoder, which incorporates supervised information into an autoencoder. The aim is to minimize the intra-class variations in the feature space and to learn discriminative manifolds of data. Em *et al.* (2017) proposed Group Sensitive TRiplet Sampling for improving the performance of fine-grained recognition by incorporating the intra-class variance structure into triplet embedding. Wang and Kosinski (2017) proposed a novel angular loss for learning the similarity metric. The proposed loss function takes the angle relationship into account and aims at constraining the angle at the negative point of triplet triangles. It is observed that the proposed algorithm is scale-invariant, captures the local structure of triplet triangles, and achieves better convergence. Wang *et al.* (2018) proposed a novel large margin cosine loss to reduce intra-class variance and maximize inter-class variance by normalization and cosine decision margin maximization.

Another popular approach for feature space optimization is using transfer learning, where the aim is to fine-tune a model trained on one task to a different but related task Pan *et al.* (2010). Jung *et al.* (2015) performed joint fine-tuning of two deep networks for facial expression recognition. The first network extracts temporal appearance features, while the second network extracts temporal geometry features. The two networks are combined with a new integration method proposed by the authors. Akata *et al.* (2015) have shown that improved performance can be achieved on unlabelled data. The authors proposed an algorithm to learn a compatibility function by assigning a higher score to matching embeddings than mismatching ones. An $l_{2,1}$ norm-based discriminative robust kernel transfer learning is proposed by Zhang *et al.* (2017) for high-level recognition tasks. Ding and Fu Ding and Fu (2017) developed a robust transfer metric learning framework by transferring the knowledge from a labeled source domain to the unlabeled target domain. Wen *et al.* (2016) proposed center loss for improving the performance of face recognition. This loss function enhances the discriminative power of the network by learning the class centers of the output features and penalizes the distance among class centers. Qi and Su (2017) proposed contrastive center loss to reduce intra-class variation and increase inter-class separability simultaneously. Kazemi *et al.* (2018) proposed attribute-centered loss for face sketch photo recognition. It incorporates attribute information in the loss function to enhance the performance of recognition. Existing literature shows that the algorithms used for feature space optimization require model training.

6.2.2 Perturbation Learning

Several attack algorithms have been proposed by researchers to showcase the vulnerability and limitation of the models. Szegedy *et al.* (2013) first discovered that deep neural networks are vulnerable to small perturbations which are imperceptible to the human eye. Goodfellow *et al.* (2014) utilize the linearity of deep neural networks in higher dimensional space and proposed fast gradient sign method to generate adversarial examples. Papernot *et al.* (2016b) showed that perturbing only 4.02% of the input results in a 97% adversarial success rate. Moosavi-Dezfooli *et al.* (2016) proposed the DeepFool algorithm to compute the accurate perturbation and evaluate the robustness of state-of-the-art classifiers. The perturbations learned using these approaches are image-

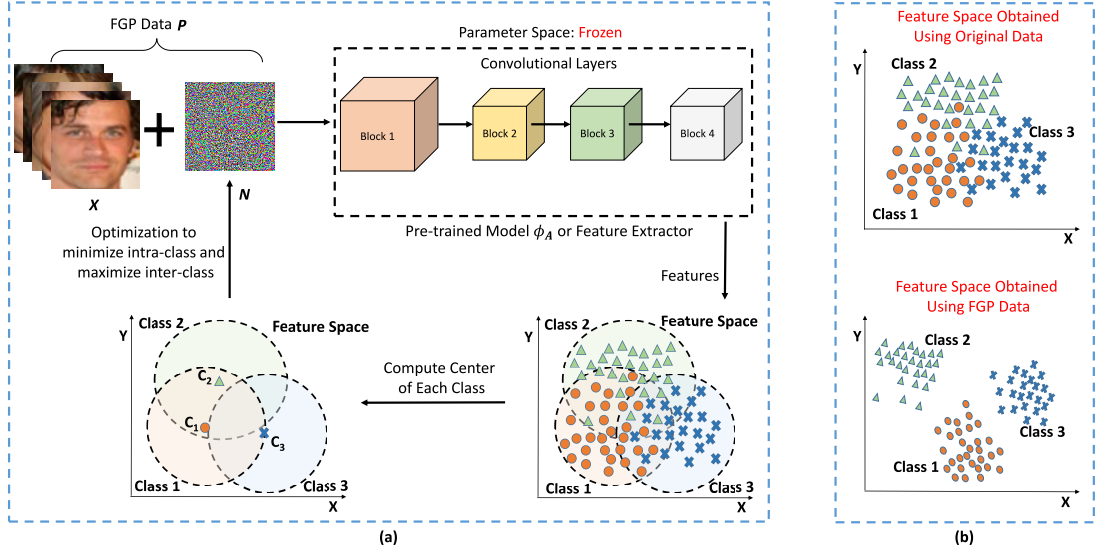


Figure 6.2: Illustration of the proposed algorithm. (a) Represents the steps involved in learning the perturbation to be added to the input data for obtaining FGP data. (b) Represents the feature space obtained using original and FGP data.

specific and require solving an optimization problem for every new image. To overcome this challenge, Moosavi-Dezfooli *et al.* (2017) proposed an image agnostic algorithm that learns a single universal perturbation corresponding to each model to fool deep neural networks.

Several researchers studied the geometrical properties of deep neural networks and later used them to develop attack and defense algorithms against adversarial attacks. Poole *et al.* (2016) used Riemannian geometry with mean-field theory to study the impact of width vs. depth on the curvature of classification decision boundary in deep networks. Later, Fawzi *et al.* (2018) used this concept and empirically showed that adversarial training reduces the curvature of the classification boundary in the input space. They also provided the analysis of the topology of decision boundary in input space. Carlini and Wagner Carlini and Wagner (2017) proposed three attacks, i.e., L_0 , L_∞ , and L_2 based on the distance metrics. Yao *et al.* (2019) claimed that the existing attack methods such as C & W are time-consuming and require many iterations with hyper-parameter tuning. This makes the learning of perturbation inefficient. Therefore, they proposed an efficient trust region-based attack method that achieves comparable results to existing attack methods and is 37x time efficient. Sayles *et al.* (2021) proposed a method to generate physical adversarial examples via making modifications in the light that illuminates the object. These physical adversarial examples have invisible perturbations. The literature shows that the applicability of perturbation learning meth-

ods beyond attack has been less explored. In this research, we have shown that these methods can be used to address the problem of facial attribute classification.

6.2.3 Facial Attribute Classification

Researchers have proposed various algorithms Mao *et al.* (2020); Zheng *et al.* (2020b); Thom and Hand (2020) for the prediction and recognition of facial attributes. The research on facial attribute classification gained impetus in 2009, with the release of the PubFig dataset by Kumar *et al.* (2009), where features extracted from facial regions are used to train a Support Vector Machine for attribute classification. In 2013, Berg and Belhumeur (2013) proposed a framework to learn discriminative intermediate-level features for a set of parts called Part-based One-vs-One Features (POOFs). Learned features are used to train SVMs for attribute classification. It is shown that POOFs do not require a large training set and can be used as an effective feature for attribute labeling in the facial domain. These methods depend on the landmark detector to detect facial landmarks. In cases of occlusion or noise, if the detector fails, then these methods would also fail. To address these challenges in unconstrained images, Liu *et al.* (2015b) proposed a deep learning-based framework. LNet and ANet are cascaded together to improve face localization and attribute prediction. Ehrlich *et al.* (2016) proposed a Multi-Task Restricted Boltzmann Machine for attribute classification by learning joint feature representation. Kalayeh *et al.* (2017) trained attribute prediction model in conjunction with a semantic segmentation model to guide the network to pay more attention to local attributes-related regions. Han *et al.* (2017a) proposed a DMTL approach to combine attribute correlation and attribute heterogeneity in a single convolutional neural network. Zhong *et al.* (2016a) have shown that CNN models trained for face recognition can be used to build facial descriptors for attribute classification. Another work by Zhong *et al.* (2016b) demonstrated that mid-level representations achieves better accuracy than a high-level abstraction for attribute classification. Hand and Chellappa (2017) proposed a Multi-Task Deep CNN for improving attribute classification by taking advantage of attribute relationships.

In 2018, Sethi *et al.* (2019) proposed a novel autoencoder-based deep learning formulation called R-Codean autoencoder for attribute prediction. He *et al.* (2018) proposed a dual-path facial attribute recognition network. The network utilizes local facial

parts and facial texture information to generate an abstraction image using the Generative Adversarial Network. Rozsa *et al.* (2019) examined the representational stability of existing algorithms by generating adversarial examples and showed that the robustness of DNNs varies between facial attributes. These methods require a large number of parameter learning and are not suitable for mobile devices. Saxen *et al.* (2019) evaluated MobileNetV2 and NasNet-Mobile lightweight CNN architecture with respect to speed and accuracy for facial attribute prediction. They observed that these two lightweight CNN architectures yield comparable performance to current state-of-the-art models and thus are suitable for mobile devices. Rose and Bourlai (2019) addressed the challenges associated with image quality. These works are dependent on global feature representation ignoring the local features at low levels for attribute prediction. To overcome this limitation, Zheng *et al.* (2020b) proposed a novel Bi-directional Ladder Attentive Network (BLAN) that exploits both global and local features for attribute prediction. A detailed survey on facial analysis is provided by Zheng *et al.* (2020a). The literature shows that these methods focused on optimizing the model and feature space for attribute prediction. However, our proposed algorithm leverage the input space to optimize the feature space for attribute prediction.

6.3 Proposed Approach

Pre-trained models generally do not yield good performance if the dataset to be evaluated is different from the pre-training dataset. In order to enhance the performance of pre-trained models on new datasets, this research proposes an algorithm to shift the data (new dataset) in the input space via perturbation learning for optimizing the feature space.

Figure 6.2(a) illustrates the proposed algorithm with an example in the 2D feature space. Here, we assume that the pre-training dataset and the dataset on which the proposed algorithm is applied contain face images. Also, the pre-training dataset is more non-linear than the dataset used for perturbation learning. Let $\mathbf{N}$ be the noise, initialized with zeros, added to input face images. The resultant perturbed images are given as input to a pre-trained model ϕ . The model ϕ outputs the feature representation of the perturbed images. Next, the center of each class is computed in the feature space.

The intra-class distance and inter-class distance are optimized over noise $\mathbf{N}$. Figure 6.2(b) shows the feature space obtained using a pre-trained model ϕ on the original input and FGP data. It can be observed that classes are overlapped in the feature space when original input data is used to extract features. However, the overlap decreases, and the compactness among the classes increases when features are extracted from the FGP data. The details of the proposed algorithm are discussed below:

Considering a pre-trained model with frozen convolutional layers as a feature extractor, the problem can be defined as “given a pre-trained model ϕ and a dataset $\mathbf{X}$, learn a single uniform perturbation $\mathbf{N}$ such that adding $\mathbf{N}$ to $\mathbf{X}$ in the input space improve the separability in the feature space”.

Let $\mathbf{X}$ be the original training set in the input space with k number of attributes, where each attribute $\mathbf{A}_j$ has t_j number of classes. Mathematically, it is represented as:

$$\mathbf{A} = \{\mathbf{A}_1(t_1), \mathbf{A}_2(t_2), \dots, \mathbf{A}_k(t_k)\} \quad (6.1)$$

Let $\mathbf{P}$ be the perturbed training set obtained after adding uniform noise $\mathbf{N}$ to the original training set $\mathbf{X}$, such that the pixel values of each perturbed image ($\mathbf{X}_i + \mathbf{N}$) is in the range of 0 to 1. In order to satisfy this constraint, $\tanh$ function is applied as follows:

$$\mathbf{P}_i = \frac{1}{2}(\tanh(\mathbf{X}_i + \mathbf{N}) + 1) \quad (6.2)$$

As mentioned above, the aim is to learn a single uniform perturbation to optimize the feature space of a pre-trained model ϕ . Let $\phi_{\mathbf{A}_j}(\mathbf{X}_i, \mathbf{W}, b)$ represent a pre-trained model with input as $\mathbf{X}_i$ and attribute $\mathbf{A}_j$, and outputs its corresponding features $\mathbf{Z}_{\mathbf{X}_i, \mathbf{A}_j}$.

$$\mathbf{Z}_{\mathbf{X}_i, \mathbf{A}_j} = \phi_{\mathbf{A}_j}(\mathbf{X}_i, \mathbf{W}, b) \quad (6.3)$$

Similarly, for any perturbed image $\mathbf{P}_i$, $\mathbf{Z}_{\mathbf{P}_i, \mathbf{A}_j} = \phi_{\mathbf{A}_j}(\mathbf{P}_i, \mathbf{W}, b)$ where, $\mathbf{Z}_{\mathbf{P}_i, \mathbf{A}_j}$ represents the features of perturbed image $\mathbf{P}_i$ with attribute $\mathbf{A}_j$. After obtaining the representation corresponding to each perturbed image $\mathbf{P}_i$, the next task is to increase the inter-class distance and decrease the intra-class distance corresponding to each attribute $\mathbf{A}_j$. The objective function is written as:

$$\min_{\mathbf{N}} \mathcal{F}(\mathbf{Z}_{\mathbf{P},\mathbf{A}}) \quad (6.4)$$

where, $\mathcal{F}(\cdot, \cdot)$ represents the function to minimize the intra-class distance and maximize the inter-class distance in the feature space. The above objective function is optimized over perturbation $\mathbf{N}$.

Let l be any class of attribute $\mathbf{A}_j$, where $l \in \{1, 2, \dots, t_j\}$ and there are h number of samples corresponding to class l in mini-batch M for attribute $\mathbf{A}_j$. To optimize the feature space, the following loss function for class l in mini-batch M is optimized:

$$\mathcal{F}(\mathbf{Z}_{\mathbf{P},\mathbf{A}_{j,l}}) = \sum_{i=1}^h \{D(\mathbf{Z}_{\mathbf{P}_i,\mathbf{A}_{j,l}}, \mathbf{C}_{A_{j,l}})\} - D(\mathbf{C}_{A_{j,l}}, \mathbf{C}_{A_{j,q}}) \quad \forall q \text{ and } l \neq q \quad (6.5)$$

where $\mathbf{Z}_{\mathbf{P},\mathbf{A}_{j,l}}$ is the feature representation of the perturbed image $\mathbf{P}_i$ with attribute $\mathbf{A}_j$ and class l . $D(\cdot)$ represents the distance metric and $\mathbf{C}_{A_{j,l}}$ and $\mathbf{C}_{A_{j,q}}$ represent the center of attribute $\mathbf{A}_j$ with class l and q , respectively. The first term minimizes the intra-class distance between the feature representation $\mathbf{Z}_{\mathbf{P}_i,\mathbf{A}_{j,l}}$ and center $\mathbf{C}_{A_{j,l}}$ of the same class. While the second term maximizes the inter-class distance between the center of class l and other class q . It is important to note that the center of each class is computed over a mini-batch M . In this research, Euclidean distance is used as the distance metric. Thus, the above equation is updated as:

$$\mathcal{F}(\mathbf{Z}_{\mathbf{P},\mathbf{A}_{j,l}}) = \max \left(0, \sum_{i=1}^h \{||(\mathbf{Z}_{\mathbf{P}_i,\mathbf{A}_{j,l}} - \mathbf{C}_{A_{j,l}})||_2\} - ||(\mathbf{C}_{A_{j,l}} - \mathbf{C}_{A_{j,q}})||_2 \right) \quad \forall q \text{ and } l \neq q \quad (6.6)$$

In order to control the separation among different classes, a margin parameter α is introduced.

$$\mathcal{F}(\mathbf{Z}_{\mathbf{P},\mathbf{A}_{j,l}}) = \max \left(0, \sum_{i=1}^h \{||(\mathbf{Z}_{\mathbf{P}_i,\mathbf{A}_{j,l}} - \mathbf{C}_{A_{j,l}})||_2\} - ||(\mathbf{C}_{A_{j,l}} - \mathbf{C}_{A_{j,q}})||_2 + \alpha \right) \quad \forall q \text{ and } l \neq q \quad (6.7)$$

Eq. 6.7 is optimized corresponding to a single class l . To optimize it for all the

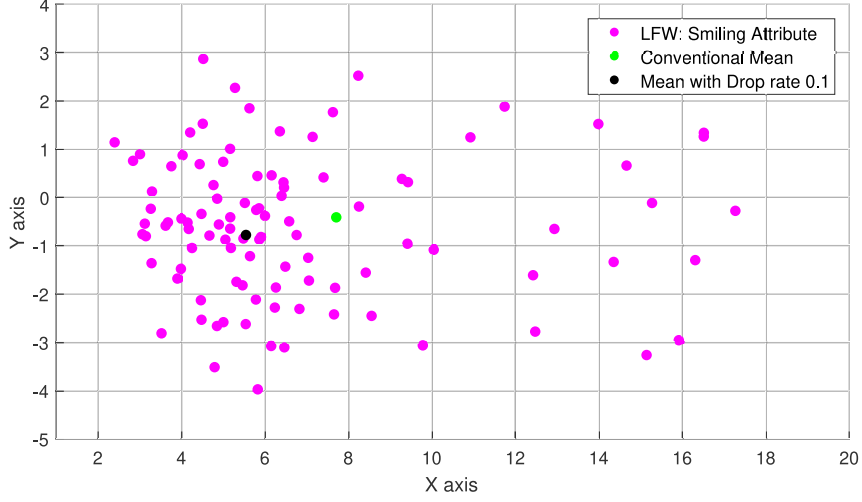


Figure 6.3: t-SNE visualization corresponding to the ‘Smiling’ attribute of the LFW dataset along with the centers computed using conventional mean and feature drop with 0.1 drop rate.

classes, it is updated as:

$$\mathcal{F}(\mathbf{Z}_{\mathbf{P}, \mathbf{A}_j}) = \sum_{l=1}^{t_j} \max\left(0, \sum_{i=1}^h \{||(\mathbf{Z}_{\mathbf{P}_i, \mathbf{A}_{j,l}} - \mathbf{C}_{A_{j,l}})||_2\} - ||(\mathbf{C}_{A_{j,l}} - \mathbf{C}_{A_{j,q}})||_2 + \alpha\right) \quad \forall q \text{ and } l \neq q \quad (6.8)$$

The above loss function optimizes the feature space via learning a perturbation $\mathbf{N}$. The learned perturbation is added to the input data for obtaining FGP data.

Feature Drop: Generally, the center of a class in feature space is computed by taking the mean of the output feature representation of that class. However, the mean can be affected by outliers present in the class. Therefore, a novel technique inspired by dropout Srivastava *et al.* (2014) is proposed to shift the center towards the denser region of a class. In conventional dropout, nodes are randomly dropped to enhance the performance of neural networks. In this research, the output feature representations in the feature space are randomly dropped to compute the center towards the denser region.

Let the probability of feature drop rate for minimizing the effect of outliers is represented by θ , which is sampled using Bernoulli distribution. As mentioned earlier, there are h samples in a class l of mini-batch M , the corresponding dropout mask vector $\mathbf{v}$ for class l is of dimension $h \times 1$ and the dropout mask for any sample i is written as $v_i \sim \text{Bernoulli}(\theta)$. The center corresponding to class l using the proposed feature drop

is computed as:

$$\mathbf{C}_{\mathbf{A}_{j,l}} = \frac{\sum_{i=1}^h v_i \mathbf{Z}_{\mathbf{P}_i, \mathbf{A}_{j,l}}}{\sum_{i=1}^h v_i} \quad (6.9)$$

Figure 6.3 shows the t-SNE visualization corresponding to the ‘Smiling’ attribute of the LFW dataset along with the centers computed using conventional mean and after applying feature drop with 0.1 drop rate. It is observed that feature drop shifts the center towards the denser region while the conventional mean is affected by outliers.

For facial attribute classification, there can be multiple cases for uniform perturbation learning, as discussed below:

Single Attribute Perturbation Learning: This case describes the scenario where a uniform perturbation is learned corresponding to a single facial attribute such as ‘Gender’. In this case $j = 1$. To optimize a uniform perturbation for a single attribute, the following loss function is optimized:

$$\begin{aligned} \mathcal{F}(\mathbf{Z}_{\mathbf{P}, \mathbf{A}_1}) = \sum_{l=1}^{t_1} \max \left(0, \sum_{i=1}^h \{ \|\mathbf{Z}_{\mathbf{P}_i, \mathbf{A}_{1,l}} - \mathbf{C}_{\mathbf{A}_{1,l}}\|_2 \} \right. \\ \left. - \sum_{q=1}^{t_1} (\|\mathbf{C}_{\mathbf{A}_{1,l}} - \mathbf{C}_{\mathbf{A}_{1,q}}\|_2 + \alpha) \right) \quad \forall q \text{ and } l \neq q \quad (6.10) \end{aligned}$$

Multiple Attributes Perturbation Learning: This case describes the scenario where a single uniform perturbation is learned for multiple facial attributes such as ‘Gender’, ‘Age’, and ‘Smiling’. For multiple attributes, the uniform perturbation is optimized using the following loss function:

$$\begin{aligned} \mathcal{F}(\mathbf{Z}_{\mathbf{P}, \mathbf{A}}) = \\ \sum_{j=1}^k \sum_{l=1}^{t_j} \max \left(0, \sum_{i=1}^h \{ \|\mathbf{Z}_{\mathbf{P}_i, \mathbf{A}_{j,l}} - \mathbf{C}_{\mathbf{A}_{j,l}}\|_2 \} \right. \\ \left. - \sum_{q=1}^{t_j} (\|\mathbf{C}_{\mathbf{A}_{j,l}} - \mathbf{C}_{\mathbf{A}_{j,q}}\|_2 + \alpha) \right) \quad \forall q \text{ and } l \neq q \quad (6.11) \end{aligned}$$

During testing, the learned perturbation is added to the test image to shift the input image to obtain the FGP data.

6.4 Experiments and Results

Data shift in the input space reduces the performance of pre-trained models. However, it is hypothesized that if the data in input space is shifted towards a region where the features are separable in feature space, then the performance of the pre-trained models is enhanced. In order to validate the hypothesis, experiments are performed using VGGFace model Parkhi *et al.* (2015). This model is pre-trained on the VGGFace dataset Parkhi *et al.* (2015) for the task of face recognition, and we have optimized its feature space for the task of facial attribute classification on three different datasets, namely LFW Huang *et al.* (2008), CelebA Liu *et al.* (2015a), and MUCT Milborrow *et al.* (2010b).

In this research, two experiments are performed (Table 6.1): (i) **Single Attribute Classification** and (ii) **Multiple Attributes Classification** and the results are compared with Data Fine-tuning (DFT) Chhabra *et al.* (2019). In the first experiment, a single perturbation is learned for each attribute, while in the second experiment, a single perturbation is learned for multiple attributes. It is important to note that the proposed algorithm does not require model training/fine-tuning to optimize the feature space.

The effectiveness of the proposed algorithm is shown by evaluating the performance of the pre-trained VGGFace model on the original and FGP data. For this purpose, features extracted using the pre-trained VGGFace model corresponding to the original and FGP data are matched using cosine distance. We termed this experiment as *Without Model Fine-tuning (MFT)*. Further, to showcase that the proposed algorithm acts as a supplement to conventional model fine-tuning to boost the overall performance, we have trained the dense/convolutional+dense layers with FGP data and compared the results with model fine-tuning on original data. Therefore, the experiments are performed under two scenarios.

1. *Model Fine-tuning (MFT) Type I: Dense Layers*: In Model Fine-tuning Type I, fine-tuning is performed by freezing the convolutional layers and updating only the dense layers (NNET)
2. *Model Fine-tuning Type (MFT) II: Convolutional Layers + Dense Layers*: In MFTType II, the last few convolutional layers along with the dense layers are updated.

Table 6.1: Summarizing the experiments performed to evaluate the proposed algorithm for facial attribute classification.

Exp.	Scenario	Dataset	Attributes
Single Attribute	MFT Type I: Dense Layers, MFT Type II: Convolutional Layers + Dense Layers	MUCT	Gender
		LFW	Smiling, Bushy Eyebrows, Pale Skin, Oval Face, Bags Under Eyes
		CelebA	Smiling, Attractive, Wearing Lipstick, Heavy Makeup, High Cheekbones
Multiple Attributes	MFT Type I: Dense Layers	LFW	{Smiling, Bags Under Eyes}, {Oval Face, Bags Under Eyes, Pale Skin}, {Bushy Eyebrows, Pale Skin}, {Smiling, Bushy Eyebrows, Pale Skin}, {Oval Face, Bags Under Eyes}
		CelebA	{Heavy Makeup, Smiling}, {Attractive, Heavy Makeup, High Cheekbones}, {High Cheekbones, Smiling}, {Attractive, Smiling, Wearing Lipstick}, {Attractive, Wearing Lipstick},
	MFT Type II: Convolutional Layers + Dense Layers	LFW	{Bushy Eyebrows, Pale Skin}, {Oval Face, Bags Under Eyes, Pale Skin}, {Smiling, Bushy Eyebrows}, {Bushy Eyebrows, Oval Face, Pale Skin}, {Oval Face, Bags Under Eyes}
		CelebA	{Attractive, Smiling}, {Heavy Makeup, High Cheekbones, Smiling} {Attractive, Wearing Lipstick}, {Heavy Makeup, High Cheekbones} {Attractive, Heavy Makeup, Wearing Lipstick},

Table 6.2: Classification accuracy (%) on five attributes obtained using original and FGP data corresponding to Without MFT.

LFW						CelebA					
Attributes	Original Data	FGP Data	FGP Data with Feature Drop			Attributes	Original Data	FGP Data	FGP Data with Feature Drop		
			0.05	0.1	0.2				0.05	0.1	0.2
Sm	61.87	83.37	83.37	81.73	84.32	Sm	55.78	80.22	81.81	81.75	81.77
BE	56.65	72.67	74.23	74.01	72.41	Att	59.07	63.01	65.17	65.27	64.97
PS	58.63	85.27	86.33	86.94	87.17	WL	67.69	77.70	81.35	82.12	79.05
OF	57.99	66.70	67.38	67.16	66.81	HM	66.03	77.65	77.36	79.26	77.52
BUE	63.05	72.90	71.53	74.01	72.67	HC	56.10	71.94	73.69	74.47	73.44

6.4.1 Datasets & Protocol

The protocol and details of each dataset are as follows:

LFW (Huang *et al.*, 2008) contains 13,143 images of 5,749 subjects. For experiments, five attributes, namely, ‘Smiling’ (Sm), ‘Bushy Eyebrows’ (BE), ‘Pale Skin’ (PS), ‘Oval Face’ (OF), and ‘Bags Under Eyes’ (BUE) are selected. The dataset is partitioned into 60% training, 20% validation, and 20% testing sets.

CelebA (Liu *et al.*, 2015a) contains a total of 202,599 face images of more than 10,000 celebrities with 40 annotated binary attributes. Five attributes, namely, ‘Smiling’ (Sm), ‘Attractive’ (Att), ‘Wearing Lipstick’ (WL), ‘Heavy Makeup’ (HM), and ‘High Cheekbones’ (HC) are randomly selected, and the standard pre-defined protocol is followed for experiments. The dataset is partitioned into 162,770 images in the training, 19,867 into validation, and 19,962 images in the testing sets.

MUCT (Milborrow *et al.*, 2010b) contains 3,755 images of 131 male and 146 female subjects. The facial region is segmented using Viola-Jones face detector Viola and Jones (2001). For each class, the dataset is partitioned into 60% training, 20% validation, and 20% testing.

Table 6.3: Classification accuracy (%) on five attributes corresponding to MFT Type I and comparison with the results of DFT Chhabra *et al.* (2019).

LFW							CelebA						
Attributes	Original Data	FGP Data	FGP with Feature Drop			DFT	Attributes	Original Data	FGP Data	FGP with Feature Drop			DFT
			0.05	0.1	0.2					0.05	0.1	0.2	
Sm	76.03	87.02	87.37	86.38	87.56	82.42	Sm	68.14	86.79	87.50	87.15	87.34	71.30
BE	68.53	80.37	81.01	80.97	79.53	69.98	Att	70.21	72.27	73.02	73.76	73.11	70.54
PS	72.11	88.66	89.69	90.07	89.69	74.81	WL	79.92	85.28	87.90	88.04	86.68	81.29
OF	68.38	74.39	74.70	74.20	73.63	69.59	HM	77.65	83.80	83.64	84.42	83.51	78.27
BUE	71.35	77.36	77.97	78.69	78.31	72.43	HC	67.70	80.07	81.23	81.50	80.63	75.38

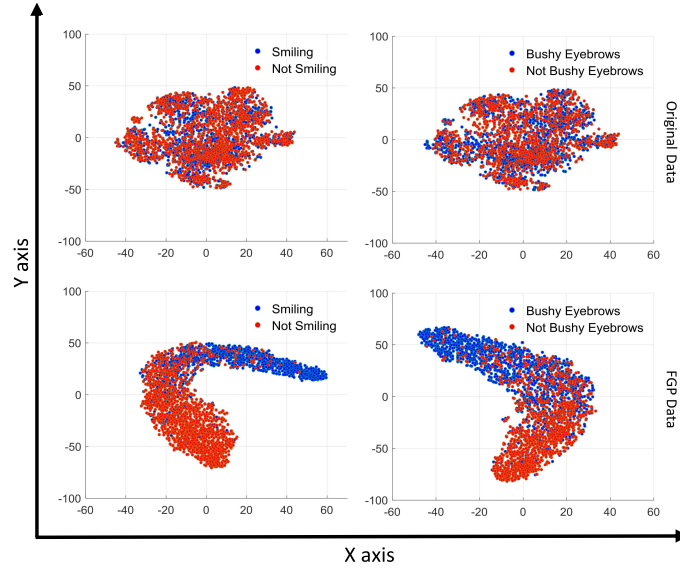


Figure 6.4: t-SNE visualization obtained using pre-trained VGGFace model pertaining to the original and FGP data corresponding to the ‘Smiling’ and ‘Bushy Eyebrows’ attributes of the LFW dataset.

6.4.2 Implementation Details

All the experiments are performed on the NVIDIA Tesla P100 server with 96GB RAM and 16GB GPU memory. Images are resized to 64×64 . Implementation details of perturbation learning and model fine-tuning are given below.

Perturbation learning: The proposed algorithm to learn a single uniform perturbation for a given dataset is implemented in Tensorflow v1.9.0. During training, the learning rate is set to 0.01, and the perturbation is trained for 800 iterations corresponding to each attribute of a given dataset. A batch size of 200 is chosen for the CelebA, LFW, and MUCT datasets. The margin α is set to 2 during the experiments.

Model Fine-tuning: Pre-trained VGGFace + Neural Network (NNET) classifier is used for both the experiments. To train NNET classifier for the MFT Type I, the learning rate is set to 0.01 for the first 10 epochs and gradually reduced by a factor of 0.1 in every 5th epoch. The model is trained using Adam optimizer for 20 epochs with a batch size

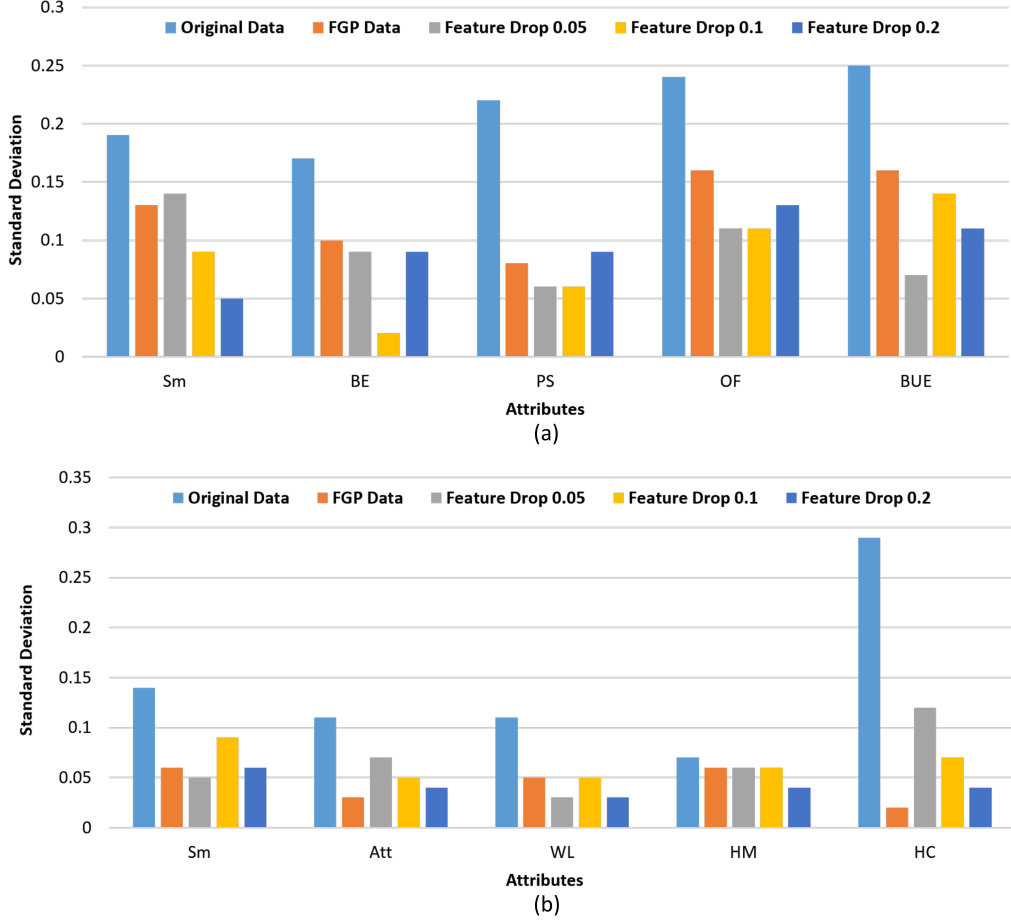


Figure 6.5: Standard deviation of five fold experiments corresponding to MFT Type I on the (a) LFW and (b) CelebA datasets.

of 32. For the MFT Type II, the last two convolutional layers along with the NNET of the pre-trained VGGFace model is fine-tuned. In this experiment, the model is trained for 20 epochs with a learning rate of 0.001 for first 10 epochs and gradually reduced by a factor of 0.1 every 5th epoch.

6.4.3 t-SNE Visualization

The t-distributed Stochastic Neighbor Embedding (t-SNE) visualization of output representation using the pre-trained model corresponding to the original and FGP data of the LFW dataset is shown in Figure 6.4. It is observed that the overlap among the classes in feature space is reduced when the FGP data is given as input to the pre-trained model. This showcases that a better feature representation of a pre-trained model can be obtained by updating the number of parameters equal to the size of the input image i.e., 12,288 ($64 \times 64 \times 3$).

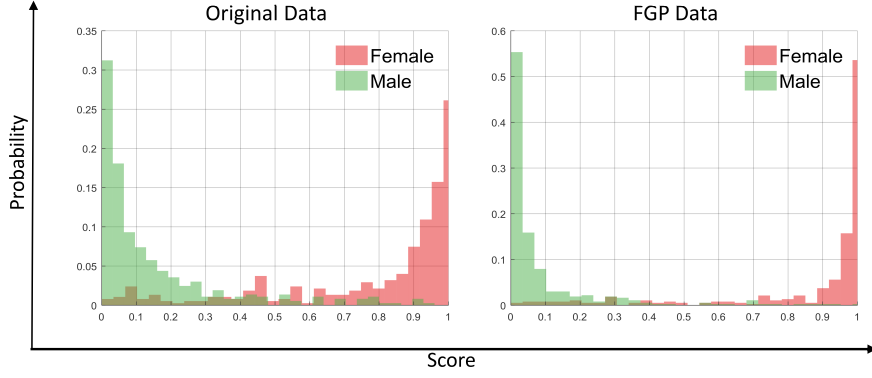


Figure 6.6: Score distribution of ‘Gender’ attribute pertaining to the original and FGP data of the MUCT dataset.

6.4.4 Single Attribute Classification

In this experiment, the performance of the proposed algorithm is evaluated for *Without MFT*, *MFT Type I*, and *MFT Type II*. Five-fold experiments are performed on the LFW, CelebA, and MUCT datasets with pre-defined protocols using original and FGP data. The results of all the experiments are discussed below.

Without Model Fine-tuning

Table 6.2 shows the comparison of the classification accuracy obtained using original and FGP data on the LFW and CelebA datasets. It is observed that the FGP data enhances the performance of the pre-trained model. For instance, the pre-trained model achieves 61.87% and 83.37% accuracy corresponding to the ‘Smiling’ attribute using original and FGP data, respectively, on the LFW dataset. Similarly, on the CelebA dataset, the model achieves an accuracy of 55.78% and 80.22%, corresponding to the ‘Smiling’ attribute using original and FGP data, respectively. This showcases that the proposed algorithm is able to enhance the performance of the pre-trained model without fine-tuning.

Model Fine-tuning Type I

In this scenario, model fine-tuning is performed by freezing the convolutional layers and updating only the dense layers. Table 6.3 shows the classification accuracy using original and FGP data on five attributes of the LFW and CelebA datasets. It is

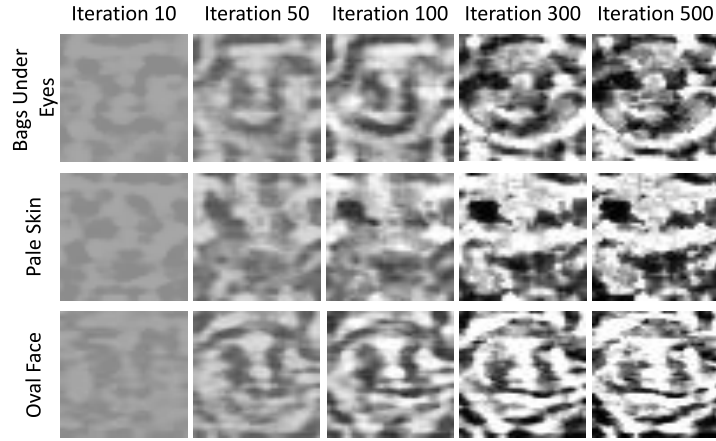


Figure 6.7: Perturbation visualization corresponding to the ‘Bags Under Eyes’, ‘Pale Skin’, and ‘Oval Face’ attributes of the LFW dataset at different iterations during training.

observed that model fine-tuning using FGP data enhances the performance for all the attributes of both datasets. For instance, the classification accuracy corresponding to the ‘Smiling’ attribute of the LFW dataset increases by approximately 11%. Similarly, the classification accuracy increases from 67.70% to 80.07% corresponding to ‘High Cheekbones’ attribute of the CelebA dataset. The significant enhancement in the classification accuracies showcases the effectiveness of the proposed algorithm. In order to further enhance the performance by minimizing the effect of outliers, Feature Drop is applied with the proposed algorithm. For this purpose, experiments are performed with drop rate $\theta = 0.05$, $\theta = 0.1$, and $\theta = 0.2$. From Table 6.3, it is observed that Feature Drop further enhances the performance by approximately 0.31% to 1.41% on the LFW dataset and 0.36% to 2.76% on the CelebA dataset for different drop rates. Figure 6.5 shows the standard deviation of the five-fold experiments on the LFW and CelebA datasets.

A comparison is also performed with data fine-tuning (DFT) (Chhabra *et al.* (2019)). It is observed that the proposed algorithm outperforms DFT for all five attributes. For instance, the classification accuracy corresponding to the ‘Smiling’ attribute of the CelebA dataset is 87.50% using FGP data with Feature Drop, which is 16.20% higher than DFT. Higher performance is achieved because the proposed algorithm shifts the data for feature space optimization.

To observe the performance of the proposed algorithm along with Feature Drop on a small dataset, the classification accuracy corresponding to the ‘Gender’ attribute

Table 6.4: Classification accuracy (%) using original and FGP data on five attributes corresponding to MFT Type II.

LFW						CelebA					
Attributes	Original Data	FGP Data	FGP Data with Feature Drop			Attributes	Original Data	FGP Data	FGP Data with Feature Drop		
			0.05	0.1	0.2				0.05	0.1	0.2
Sm	87.25	88.70	88.77	88.20	88.05	Sm	78.43	88.54	88.7	88.90	88.65
BE	76.33	82.08	82.08	81.66	81.70	Att	71.37	75.23	74.65	74.92	74.69
PS	74.85	90.37	90.30	90.30	90.98	WL	84.88	88.67	89.44	89.50	88.76
OF	70.40	76.45	76.18	76.60	74.89	HM	81.02	85.32	85.55	85.84	85.24
BUE	74.81	78.58	78.16	78.92	79.30	HC	72.56	82.54	83.06	83.43	82.80

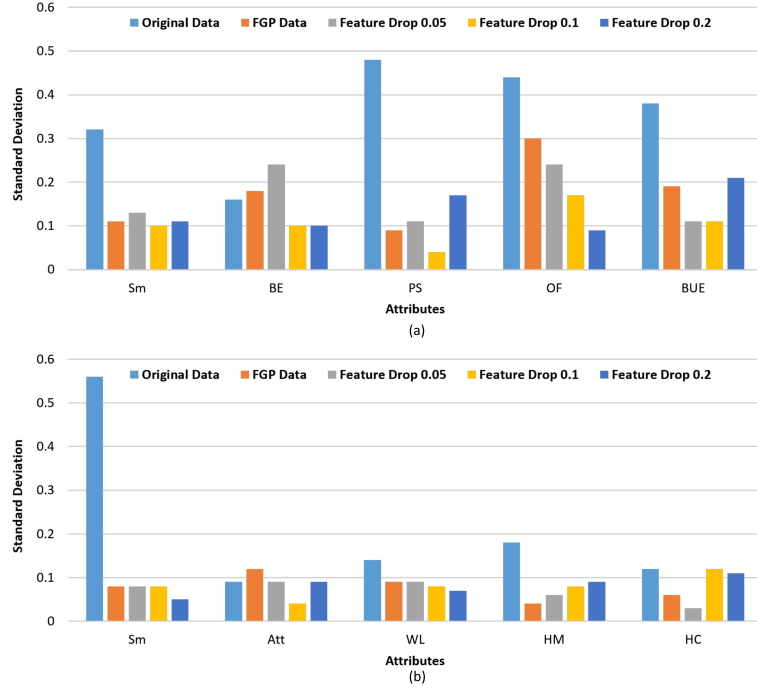


Figure 6.8: Standard deviation of five fold experiments corresponding to MFT Type II on the (a) LFW and (b) CelebA datasets.

of the MUCT dataset is also computed. The accuracy corresponding to this attribute increases from 87.57% to 94.19%, and the standard deviation reduces from 0.17% to 0.08% using FGP data. While using Feature Drop with 0.05, 0.1, and 0.2 drop rates, a classification accuracy of 93.78%, 94.05%, and 94.32% along with standard deviation of 0.09%, 0.06%, and 0.07% is achieved. Figure 6.6 shows the output score distribution corresponding to the original and FGP data of the ‘Gender’ attribute. It is observed that model fine-tuning using FGP data reduces the overlap between the classes and increases the confidence of predicting the true class scores. These experiments show the generalizability of the proposed algorithm for small as well as large-scale datasets. Figure 6.7 shows the perturbation visualization corresponding to ‘Bags Under Eyes’, ‘Pale Skin’, and ‘Oval Face’ attributes of the LFW dataset at different iterations during training. It is observed that the model focuses on attribute-specific regions with increasing iterations.

Table 6.5: Comparing the number of parameters and computational time with mean classification accuracy (%) corresponding to the five attributes of the CelebA and LFW datasets using the original and FGP data.

Model Fine-tuning	Number of Parameters	Time in Sec.		Mean Accuracy	
		CelebA	LFW	CelebA	LFW
Type II (Original Data)	5.25M	145.00	6.00	77.65	76.73
Type I (FGP Data)	0.54M	114.05	5.04	81.64	81.56

Table 6.6: Comparing the performance with pre-trained VGGFace model on multiple attributes (combination of two and three attributes).

Two Attributes						
LFW						
Attributes	Sm	BUE	BE	PS	OF	BUE
Original Data	61.87	63.05	56.65	58.63	57.99	63.05
FGP Data	78.57	66.93	63.81	78.76	60.54	66.93
CelebA						
Attributes	HM	Sm	Att	WL	HC	Sm
Original Data	66.03	55.78	59.07	67.69	56.10	55.78
FGP Data	71.76	69.06	63.00	73.63	70.98	78.34
Three Attributes						
LFW						
Attributes	Sm	BE	PS	OF	BUE	PS
Original Data	61.87	56.65	58.63	57.99	63.05	58.63
FGP Data	68.60	62.36	74.92	56.50	64.23	72.75
CelebA						
Attributes	Att	HM	HC	Att	Sm	WL
Original Data	59.07	66.03	56.10	59.07	55.78	67.69
FGP Data	63.75	72.58	63.21	63.30	65.90	72.99

Model Fine-tuning Type II

In this scenario, the pre-trained model is fine-tuned by training the last two convolutional layers along with the dense layers corresponding to the LFW and CelebA datasets. For the MUCT dataset, only the last convolutional layer is trained along with the dense layers. The classification accuracy using the fine-tuned models corresponding to the original and FGP data of the LFW and CelebA datasets is shown in Table 6.4. Similar to MFT Type I, the classification performance is improved, corresponding to all the attributes of both datasets. For instance, the classification accuracy corresponding to ‘Bushy Eyebrows’ and ‘Pale Skin’ attributes of the LFW dataset increases by 5.75% and

Table 6.7: Classification accuracy (%) on multiple attributes (combination of two and three attributes) corresponding to MFT Type I.

Two Attributes						
LFW						
Attributes	Sm	BUE	BE	PS	OF	BUE
Original Data	76.03	71.35	68.53	72.11	68.38	71.35
FGP Data	84.25	75.11	75.46	86.34	70.43	74.81
CelebA						
Attributes	HM	Sm	Att	WL	HC	Sm
Original Data	77.65	68.14	70.21	79.92	67.70	68.14
FGP Data	80.41	80.09	72.39	82.87	79.45	85.27
Three Attributes						
LFW						
Attributes	Sm	BE	PS	OF	BUE	PS
Original Data	76.03	68.53	72.11	68.38	71.35	72.11
FGP Data	80.94	74.32	83.60	67.28	72.22	83.87
CelebA						
Attributes	Att	HM	HC	Att	Sm	WL
Original Data	70.21	77.65	67.70	70.21	68.14	79.92
FGP Data	71.20	79.63	73.90	71.81	78.11	81.60

15.52%, respectively. Similarly, the classification accuracy corresponding to the ‘Gender’ attribute of the MUCT dataset increases from 94.32% to 96.08%, and the standard deviation decreases from 0.12% to 0.07%. This demonstrates that the data shift using the proposed algorithm acts as a supplement for traditional model fine-tuning. Similar to MFT Type I, Feature Drop is applied with the proposed algorithm with the same drop rates. It is observed that Feature Drop marginally improves the performance. The standard deviation of the five-fold experiments on the LFW and CelebA datasets are shown in Figure 6.8.

Apart from this, a comparison of the number of parameters updated during MFT Type II (conv + dense layers) using original data and MFT Type I (dense layers) using FGP data is performed. Table 6.5 shows the classification accuracy for the same. It is observed that model fine-tuning using original data results in 5.25M parameters (conv + dense layers) update and achieves a mean classification accuracy of 77.65% and 76.73% corresponding to the CelebA and LFW datasets, respectively. However, model fine-tuning using FGP data achieves better performance by only updating 0.54M parameters (dense layers + perturbation). We have also compared the computational time. It is found that model fine-tuning using original data takes 145.00 seconds for one epoch (conv + dense layers) on the CelebA dataset, while MFT using FGP data takes only

Table 6.8: Classification accuracy (%) on multiple attributes (combination of two three attributes) corresponding to MFT Type II.

Two Attributes						
LFW						
Attributes	BE	PS	OF	BUE	Sm	BE
Original Data	76.33	74.85	70.40	74.81	87.25	76.33
FGP Data	79.07	89.00	74.12	76.90	87.60	81.85
CelebA						
Attributes	Att	WL	HM	HC	Att	Sm
Original Data	71.37	84.88	81.02	72.56	71.37	78.43
FGP Data	74.56	87.41	83.13	81.07	74.05	87.02
Three Attributes						
LFW						
Attributes	OF	BUE	PS	BE	OF	PS
Original Data	70.40	74.81	74.85	76.33	70.40	74.85
FGP Data	71.16	75.88	88.01	78.20	70.17	87.40
CelebA						
Attributes	Att	HM	WL	HM	HC	Sm
Original Data	71.37	81.02	84.88	81.02	72.56	78.43
FGP Data	74.83	84.44	87.80	83.51	81.66	86.85

114.05 seconds for one epoch (dense layers + perturbation).

6.4.5 Multiple Attributes Classification

In this experiment, the combination of attributes is taken together for perturbation learning. As shown in Table 6.1, the performance of the proposed algorithm for multiple attributes is evaluated on a combination of two and three attributes from the LFW and CelebA datasets for *Without MFT* and MFT Type I and *Type II*. Similar to Single Attribute Classification, the proposed algorithm is evaluated for *Without MFT*, *MFT Type I*, and *MFT Type II*.

Without MFT

The classification accuracy obtained using original and FGP data on the LFW and CelebA datasets is shown in Table 6.6. The results showcase that the proposed algorithm enhances the performance of the pre-trained model using FGP data. For instance, the accuracy for the combination of {'High Cheekbones', 'Smiling'} of the CelebA dataset increases from 56.10% and 55.78% to 70.98% and 78.34%.

Table 6.9: Comparison of the classification accuracy (%) of the proposed algorithm with FLDA on the LFW and CelebA datasets.

LFW			CelebA		
Attributes	FLDA	FGP	Attributes	FLDA	FGP
Sm	75.72	87.02	Sm	66.01	86.79
BE	67.69	80.37	Att	68.93	72.27
PS	71.91	77.36	WL	76.50	85.28
OF	67.31	88.66	HM	75.16	83.80
BUE	71.57	74.39	HC	65.44	80.07

Table 6.10: Analyzing the effect of varying batch size at learning rate = 0.01 and $\alpha = 2.0$ on the LFW dataset.

Attributes	MFT: Type I			
	50	100	200	300
Sm	85.19	87.29	87.02	86.83
BE	79.22	79.37	80.37	81.12
PS	87.29	87.82	88.66	88.58
OF	70.28	72.14	74.39	73.63
BUE	75.19	75.98	77.36	77.09
Attributes	MFT: Type II			
	50	100	200	300
Sm	87.13	89.11	88.70	88.50
BE	81.27	81.27	82.08	82.61
PS	89.57	89.42	90.37	89.65
OF	73.85	75.41	76.45	75.49
BUE	76.82	76.97	78.58	78.23

Table 6.11: Analyzing the effect of varying learning rate at batch size = 200 and $\alpha = 2.0$ on the LFW dataset.

Attributes	MFT: Type I			
	0.0001	0.005	0.001	0.01
Sm	79.64	87.40	82.95	87.02
BE	69.14	79.71	72.41	80.37
PS	73.97	87.17	81.27	88.66
OF	69.74	73.21	68.49	74.39
BUE	72.37	76.67	73.89	77.36

Attributes	MFT: Type II			
	0.0001	0.005	0.001	0.01
Sm	88.24	88.88	87.59	88.70
BE	77.58	81.20	77.35	82.08
PS	77.54	89.19	83.44	90.37
OF	74.35	74.84	72.90	76.45
BUE	77.24	78.69	77.35	78.58

Model Fine-tuning Type I and Type II

Table 6.7 shows the classification accuracy using original and FGP data for the combination of two and three attributes, respectively, corresponding to MFT Type I on the LFW and CelebA datasets. It is observed that model fine-tuning using FGP data outperforms for all the combinations of attributes. For instance, the classification accuracy for the combination of {‘Bushy Eyebrows’, ‘Pale Skin’} of the LFW dataset increases from 68.53% and 72.11% to 75.46% and 86.34%. For the CelebA dataset, FGP data enhances the performance by 11.75% and 17.72% for the combination of {‘High Cheekbones’, ‘Smiling’}. A similar performance boost is observed for the combination of three attributes for both datasets. The results for MFT Type II are shown in Table 6.8. It is found that FGP data enhances the classification performance for MFT Type II as well. The enhanced performance showcases the capability of the proposed algorithm in learning better feature representation. It is also observed that on increasing the number of attributes, the performance of the proposed algorithm decreases. It is our assertion that on increasing the number of facial attributes, the joint optimization by learning a

Table 6.12: Analyzing the effect of varying α at batch size = 200 and learning rate = 2.0 on the LFW dataset.

Attributes	MFT: Type I			
	1.0	1.5	2.0	2.5
Sm	86.11	87.90	87.02	87.10
BE	79.79	79.03	80.37	79.76
PS	88.77	88.50	88.66	88.01
OF	73.17	72.95	74.39	72.98
BUE	77.02	76.41	77.36	76.14

Attributes	MFT: Type II			
	1.0	1.5	2.0	2.5
Sm	88.77	87.76	88.70	89.04
BE	80.82	79.79	82.08	80.93
PS	90.47	89.46	90.37	89.88
OF	75.45	75.49	76.45	74.24
BUE	79.19	78.31	78.58	78.31

single uniform perturbation becomes difficult.

In order to showcase the effectiveness of the proposed algorithm, results are computed on all the attributes of both LFW and CelebA datasets. The performance of the proposed algorithm is compared with existing algorithms. The average classification accuracy corresponding to the LFW and CelebA datasets are 86.30% and 90.90%, respectively. Figure 6.9 shows the bar graph for the comparison with state-of-the-art algorithms for the LFW dataset. The proposed algorithm achieves second-best performance. Existing algorithms primarily focus on designing different deep model architectures and feature extraction techniques for attribute classification. For instance, Hand and Chellappa Hand and Chellappa (2017) proposed a Multi-Task Deep CNN for improving attribute classification by taking advantage of attribute relationships. Similarly, a dual-path facial attribute recognition network is proposed by He *et al.* (2018). On the other hand, Zhong *et al.* (2016b) proposed to leverage mid-level feature representation from deep CNN for attribute classification. However, the proposed algorithm is independent of network architecture and the feature extraction technique.

Table 6.13: Classification accuracy (%) of the proposed algorithm with feature drop (best performance), median, and interquartile range corresponding to MFT: Type I and MFT: Type II on the LFW dataset and CelebA dataset.

MFT: Type I	LFW			
	Attributes	Feature Drop	Median	Inter-quartile
	Sm	87.56	85.08	84.70
	BE	81.01	77.51	73.82
	PS	90.07	83.40	81.54
	OF	74.70	71.88	71.08
	BUE	78.69	74.20	73.59
	CelebA			
	Sm	87.50	78.59	74.18
	Att	73.76	72.44	70.45
	WL	88.04	83.00	80.86
	HM	84.42	79.65	78.64
	HC	81.50	75.14	72.34
MFT: Type II	LFW			
	Sm	88.77	87.59	87.59
	BE	82.08	80.86	78.50
	PS	90.98	86.45	84.09
	OF	76.60	73.28	75.38
	BUE	79.30	76.21	76.06
	CelebA			
	Sm	88.90	85.27	83.89
	Att	74.92	75.01	74.87
	WL	89.50	87.41	86.58
	HM	85.84	83.75	82.82
	HC	83.43	80.74	79.26

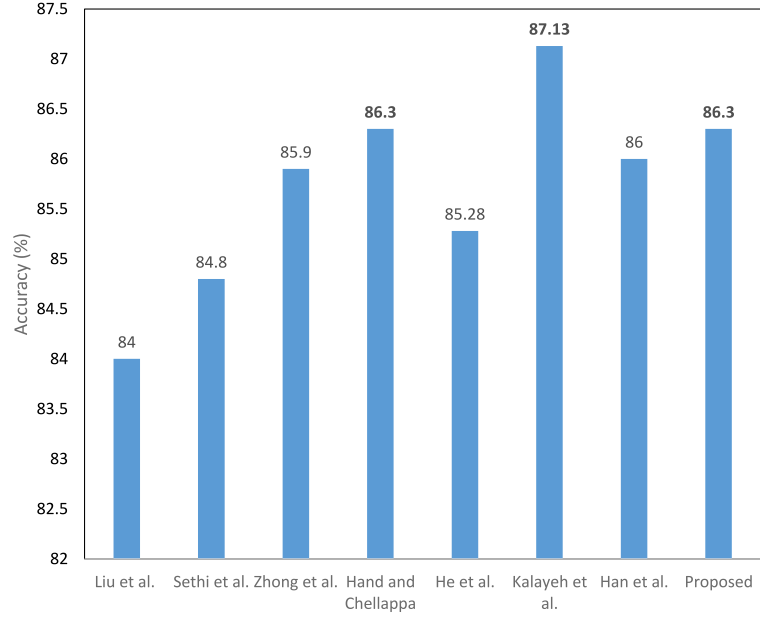


Figure 6.9: Summarizing the results of the proposed algorithm and state-of-the-art algorithm on the LFW dataset with 40 attributes.

6.4.6 Comparison with Existing Algorithms

The proposed algorithm shifts the data through a learned linear operation, i.e., adding (learned) perturbation noise N to the dataset X , by invoking the Fisher criterion. Therefore, we have compared the proposed algorithm with Fisher Linear Discriminant Analysis (FLDA) as a baseline comparison. The results are summarized in Table 6.9 shows that the proposed algorithm outperforms FLDA for all the attributes on both datasets. We would like to mention that the proposed algorithm is a learning algorithm that learns or captures the shifts required in the input space such that the output feature space is optimized. For optimizing the feature space, we use a loss function that provides supervision in the input space to learn perturbation by minimizing the intra-class distance and maximizing the inter-class distance in the feature space. It is our assertion that the low performance of FLDA is due to dimensionality reduction while projecting the data linearly. Dimensionality reduction leads to information loss, which in turn affects the overall performance.

6.4.7 Choice of Hyperparameters

To analyze the effect of hyperparameters while perturbation learning, we have performed experiments by varying the batch size, learning rate, and α . The results are

reported in Tables 6.10, 6.11, and 6.12. It is observed that the classification performance decreases when the perturbation is learned using a small batch size. For instance, the classification accuracy decreases from 74.39% to 70.28% for the ‘Oval Face’ attribute when the batch size is reduced from 200 to 50, corresponding to *MFT Type I*. Similarly, on decreasing the learning rate, it is found that the classification accuracy decreases. On analyzing the effect of varying α , it is found that the best classification performance is achieved at $\alpha = 2.0$ for the majority of the attributes.

6.4.8 Effectiveness of Feature Drop

The objective of using the feature drop method with the proposed algorithm is to minimize the effect of outliers and move the center towards the denser region. In order to analyze the effect of the feature drop method, we have compared the performance of the feature drop technique with median and interquartile ranges. Table 6.13 shows the comparison of results corresponding to both MFT Type I and Type II, respectively. It is observed that the proposed feature drop technique outperforms other approaches.

6.4.9 Robustness against New Biases and Distortions

In FGP, safeguards against biases and distortions are built into the feature-level optimization. Perturbations are guided by feature embeddings to cluster same class samples and separate different classes, while a Feature Drop mechanism discards outliers when computing class centers. This prevents the perturbation from being driven by anomalous or biased samples, ensuring it reflects the dense, representative structure of the class. Like DFT, the perturbation is applied uniformly across the dataset, avoiding sample-specific distortions. Evaluations across CelebA, LFW, and MUCT demonstrate that improvements generalize across demographic attributes, providing evidence that the proposed method enhances robustness and separability in feature space without introducing spurious correlations or dataset-specific artifacts. Together, these safeguards ensure that performance gains are achieved while maintaining fairness, semantic integrity, and generalization.

6.5 Summary

Predicting facial attributes plays a key role in various applications ranging from targeted advertisements to security and surveillance. It has inspired many industries and government organizations to develop and deploy sophisticated systems for facial attribute classification. However, these models generally do not perform well in real-world scenarios when there is a data shift between training and testing sets. In this research, we present a novel algorithm to address data shift by shifting the data in input space to obtain Feature Guided Perturbed (FGP) data by adding a uniform perturbation to the input data. Perturbation is learned by bringing the samples of the same class towards its class center and pushing away from other class centers in the feature space. Traditionally, perturbations are added to reduce the performance, whereas the proposed algorithm incorporates it for performance improvement. Another major contribution of this research is to minimize the effect of outliers while computing class centers in feature space using the proposed Feature Drop method. The proposed method uses a hyperparameter p to randomly drop features. Multiple experiments are performed to evaluate the efficacy of the proposed algorithm for single and multiple attribute classification. It is observed that classes are better separated in the feature space by adding the learned uniform perturbation, thereby improving the overall classification performance.

Chapter 7

DIVINE: Diverse-Inconspicuous Feature Learning to Mitigate Abridge Learning

7.1 Introduction

Deep learning algorithms have achieved tremendous success in several tasks including image classification (Taesiri *et al.*, 2024), (Dehghani *et al.*, 2023), (Su *et al.*, 2023), object detection (Wang *et al.*, 2023b), (Zong *et al.*, 2022), (Tan *et al.*, 2020), and segmentation (Fang *et al.*, 2023), (Xie *et al.*, 2020). However, the robustness and generalizability of these algorithms in real-world scenarios is still an open problem. Supervised learning tasks in deep neural networks primarily focus on maximizing the classification accuracy by learning the easiest solutions/patterns that exist in the entire dataset (Geirhos *et al.*, 2020). In other words, models take *shortcuts* by learning only the dominant input features that are sufficient for confident classification (Dogra *et al.*, 2024), (Li *et al.*, 2023c), (Ilyas *et al.*, 2019), (Pezeshki *et al.*, 2021). This results in ignorance of other useful and distinct features that can be helpful in classification. Therefore, as shown in Figure 7.1, these models often fail to classify *out-of-distribution* samples.

In this research, we termed the above-mentioned learning process as “Abridge Learning”. Formally, Abridge Learning (AL) is defined as the “*learning process in which a model learns only the dominant input features while failing to learn other useful input features for the target task*”. The solution obtained by Abridge learning process lacks generalizability and is not suitable for deployment in real-world scenarios. To mitigate the problem of Abridge Learning, one solution is to identify the inconspicuous discriminative input features and use them to learn a diverse unified model that is generalizable to unseen real-world datasets. In an ideal training process, a model should identify and learn all the input features present; however, the identification of all the input features is intractable. Therefore, it is important to identify a set of diverse input features that can provide a generalized solution.

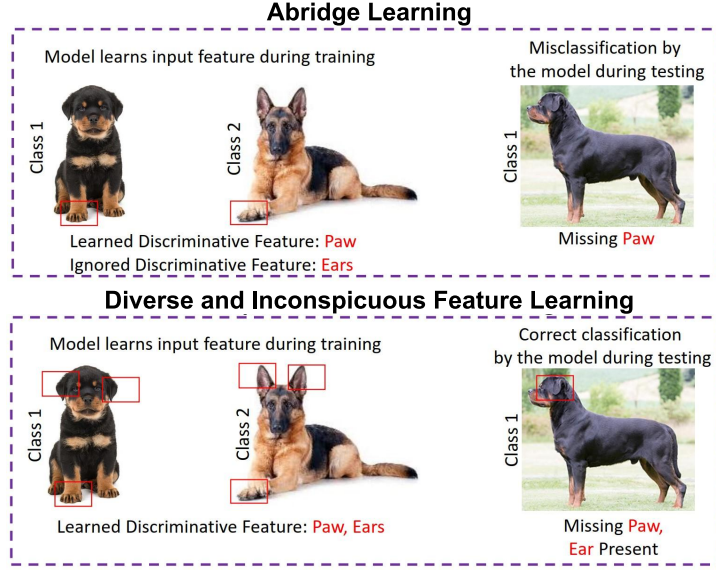


Figure 7.1: Illustration of the *Abridge Learning* (AL) and the proposed method. Model obtained with AL learns only ‘paw’ input feature and ignore other features. This results in failure of the model on the image with missing ‘paw’ feature. The Proposed DIVINE learns ‘paw’ along with feature ‘ear’ as well. This results in successful classification of the image where ‘paw’ feature is missing and the ‘ear’ feature is present.

Research Contributions: Existing methods address the problem of robustness and lack of generalizability in models; however they do not explicitly focus on learning the inconspicuous features from the dataset that are also discriminative in nature. This leads to learning of shortcut features that only work well for a given task on the given data but cannot sustain under real-world variations in data. To abridge this gap, we propose a novel learning method termed as DIVINE i.e., Diverse-Inconspicuous Feature Learning which mitigates this problem by removing the shortcuts and learning a diverse set of input features. The objective of the proposed method is two-fold: (a) identification of the minimum number of diverse inconspicuous discriminative input features as illustrated in Figure 7.1 and (b) train a unified model to learn the identified features for enhancing generalizability. The diverse features identified using the proposed method are disjoint, which in turn maximizes diverse learning of the model. Experiments for image classification tasks on CIFAR10, MNIST, CIFAR10-C, and CIFAR100-C datasets show that, DIVINE is generalized and can be applied to different machine learning tasks.

7.2 Related Work

The notion of “Abridge Learning” is excessively applicable to train deep learning models. This learning schema has been termed as shortcut learning¹(Geirhos *et al.*, 2020) where the model finds shortcuts to minimize the loss, primarily by picking up only the dominant features in the input. This has been the traditional way of training the models. Carter *et al.* (2021) have shown that only 5% spurious pixel subsets are enough for confident prediction leading to over-interpretation of pixels resulting in shortcut learning. Such shortcuts were also recently identified in a medical imaging task (Oakden-Rayner *et al.*, 2020) where the model picked undesired tumor patterns in the dataset. It gave a falsely reasonable performance, which went undetected because of the chosen evaluation metric. This implies that such spurious correlations can also remain unobserved because of the selected evaluation metrics. Lapuschkin *et al.* (2019) discussed Clever Hans strategies in the domain of Computer Vision and Arcade Gaming. They generated heatmaps and showed the patterns focused by the model while taking shortcuts.

Various drawbacks of shortcut learning process have recently been identified and addressed by the community. One such learning process termed as Gradient Starvation (Pezeshki *et al.*, 2021) showed that the deep learning model extracts a subset of features to minimize the loss for training, while the gradients from other potentially essential features starve. Du *et al.* (2020) proposed a CREX method to avoid shortcuts. This method regularizes the subset of features annotated by experts. Wang *et al.* (2023a) proposed DFM-X, which use the prior knowledge from the previous models to train the target model to enhance the generalizability and robustness of the model. Zhang *et al.* (2023) proposed SADA method to generate images for the data augmentation corresponding to highly sensitive frequencies. Gao *et al.* (2023) proposed DDA based adaptation method, which learns the diffusion model on the source data and projects the target input to the source during testing. Guo *et al.* (2023) proposed a method to reduce the sensitivity of vision transformer against patch based corruptions. Ross and Doshi-Velez (2018) found that the networks trained with input gradients are more robust and generalizable. A similar approach to make models more robust involves regularizing the gradient norm of the model output with respect to the inputs. These Jacobian-based regularizers have been shown to significantly improve classification accuracy (Varga *et al.*, 2017). Another approach incorporating the Jacobian in deep models is Jacobian

Adversarially Regularized Networks (JARN), where the model’s Jacobian is optimized by adversarial regularization (Chan *et al.*, 2020). Similar insights have been utilized to evaluate the robustness of deep models on data containing random and adversarial input perturbations (Hoffman *et al.*, 2019). The input perturbations and corruptions in the data are widely studied in the literature to evaluate the robustness of models (Rusak *et al.*, 2020) (Taori *et al.*, 2020). Different approaches (Bai *et al.*, 2021; Strisciuglio and Azzopardi, 2022; Krueger *et al.*, 2021; Benkert *et al.*, 2022; Machireddy *et al.*, 2022; Timpl *et al.*, 2022; Chefer *et al.*, 2022) are used to mitigate the effect of such distribution shifts. For example, Hendrycks *et al.* (2021) proposed a data augmentation method to address this problem, which includes geographic location as well as camera operation. Another work discusses the impact of manipulating batch normalization statistics for corrupted data to improve model performance (Schneider *et al.*, 2020).

Several data augmentation based methods have been proposed in the literature to directly or indirectly address the problem of shortcut learning Jin *et al.* (2024). DeVries (2017) proposed a method CutOut which randomly mask out the square regions in the input during model training to improve the model robustness. Zhong *et al.* (2020) extended the work of DeVries (2017) and assigns the random pixel value to the masked region to improve model performance. Cutmix replaces the region of an image with a region of different image (Yun *et al.*, 2019). Zhang *et al.* (2018) proposed similar work, where the model is trained in convex combination of pairs of images and their corresponding labels rather than replacing a portion of one image with portion of another image. Hendrycks *et al.* (2019) proposed data augmentation technique AugMix which utilizes stochasticity and diverse augmentations to produce a high diversity of augmented images to improve model robustness. Huang *et al.* (2023) proposed a simple data augmentation IPMix, which integrates pixel-based, patch-based, and image-based data augmentation into a single framework to achieve model robustness. NoisyMix proposed by Erichson *et al.* (2024) used noisy augmentations in both input and feature space to improve model robustness. To address the problem of label ambiguities incurred during data augmentation with combining multiple images, Islam *et al.* (2024a) proposed DIFFUSEMIX method which leverages stable diffusion for data augmentation. Authors have extended their work and proposed GenMix (Islam *et al.*, 2024b) a prompt guided data augmentation method to enhance model robustness.

Recently, Li et al. Li *et al.* (2023c) proposed a Last Layer Ensemble (LLE) method

to mitigate the multiple shortcuts present in the dataset. The authors stated in the paper that the previous methods were amplifying the reliance of the models on other shortcuts while mitigating one shortcut and their proposed method is able to mitigate the multiple shortcuts. However, these shortcuts should be prior known for mitigation. Yang *et al.* (2024b) made multiple contributions to identify the mitigate spurious correlations. The authors showed that the gradient based methods can be used to identify spurious correlations in the initial training phase. The authors proposed SePARate early and REsample (Spare) method, which leverage importance sampling method to mitigate shortcuts. Zhao *et al.* (2024) proposed a method COMI (CORrect and MItigate) to address the problem of shortcut learning. The proposed method reduces the dependency of models on shortcuts and extracts meaningful features integrated with standard Empirical Risk Minimization (ERM). Deng *et al.* (2024) showed that the imbalanced data group are more easily learned rather than core feature and overshadows the learning of core features by spurious feature. The authors proposed Progressive Data Expansion (PDE) training algorithm, which identifies and uses the group information to improve the model’s robustness against shortcuts. Other approaches (Ling *et al.*, 2024), (Kwon *et al.*, 2024) mitigate shortcuts for specific applications such privacy preservation and semantic segmentation.

7.3 Diverse-Inconspicuous Feature Learning

As shown in Figure 7.2, the proposed DIVINE algorithm is a two-part learning approach. In the first learning part, the proposed algorithm identifies and then suppresses the learned input features via a dominance feature map to identify other inconspicuous input features. The dominance feature map represents the dominance of the identified input features for classification. Further, in the second part, a unified model is trained using these dominance feature maps to learn all the identified input features for enhancing generalizability. It should be noted that the problem of identification of all the inconspicuous features is intractable, and hence, the proposed method identifies a set of diverse inconspicuous features that maximize the learning of the model. As a result,

¹In the literature, Shortcut Learning is referred to as the broad term for various ways in which a model takes shortcuts by learning unintended strategies to minimize the loss. Abridge Learning is a sub-problem of Shortcut Learning that deals only with the problem where the model picks up only dominant cues and ignores other relevant features from the input data.

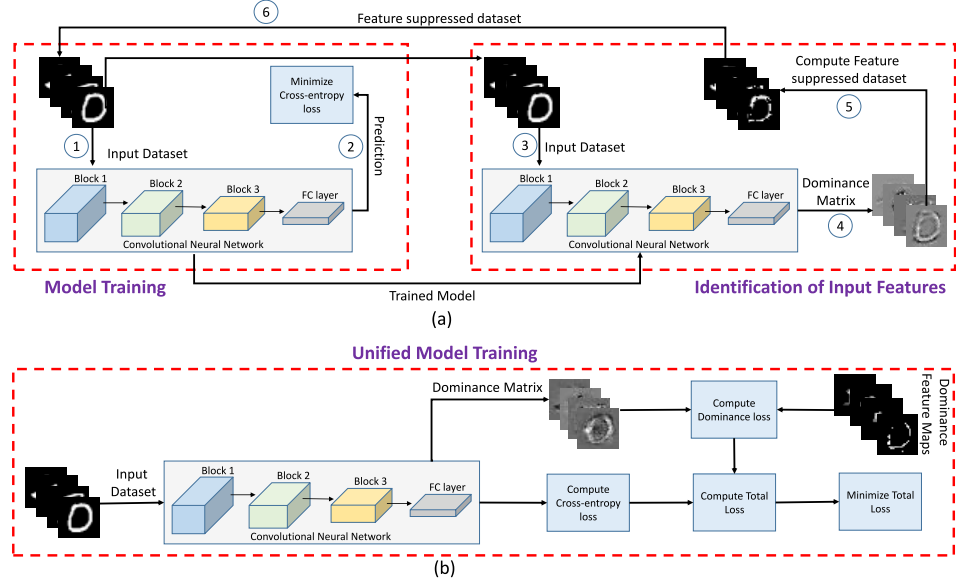


Figure 7.2: Pipeline of the proposed diverse and inconspicuous feature learning method. (a) Shows the steps to identify the inconspicuous input features. In steps 1 and 2, the model is trained with original images by minimizing cross-entropy loss function. Steps 3 and 4 shows the computation of the dominance matrix corresponding to each image. In step 5, feature-suppressed dataset is obtained via the dominance matrix and dominance feature maps. In the final step, the feature-suppressed dataset is used for training the model to identify the inconspicuous input features. (b) Shows the unified diverse model training with original images.

the proposed algorithm is able to alleviate the problem of Abridged Learning, and the generalizability towards the out-of-distribution data samples.

Let $\mathbf{X} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ be a dataset with original training images and their corresponding labels. For simplicity, let x is an image of the dataset $\mathbf{X}$ with label y in one-hot form. Consider a model $f_{\mathbf{X}}(\cdot; \theta)$ with parameters θ . This model is trained on the dataset $\mathbf{X}$ using cross entropy loss function, optimized over parameters θ .

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathbf{X}} [-y^T \log f_{\mathbf{X}}(x; \theta)] \quad (7.1)$$

where, $f_{\mathbf{X}}(x; \theta)$ outputs the probability vector for a given input image x . As mentioned earlier, the objective is to identify diverse inconspicuous input features via dominance feature maps followed by training a unified diverse model for enhancing generalizability. Here, we will first discuss the method for identification of inconspicuous features followed by training of the unified model.

7.3.1 Identification of Inconspicuous Input Features

Let $\mathbf{F}(\mathbf{x}) = \{F_1(x), F_2(x), \dots, F_r(x)\}$ be the set of all input features present in the image x that can be learned by the model using the loss function in Eq. 7.1. Each feature $F_i(x)$ in the feature set $\mathbf{F}$ is the combination of input image pixels that can be learned by the model for classification. For example, in Figure 7.1, one of the image features (in the feature set learned by the model) constitutes of pixels of the *paw* region. From the literature (Geirhos *et al.*, 2020) (Pezeshki *et al.*, 2021), it is known that during training, a model learns the most dominant input features present in the dataset. Hence, the first step is to identify the dominance of each pixel in an input image x for classification. For this purpose, we have used input-output Jacobian method (Chan *et al.*, 2020) (Hoffman *et al.*, 2019) that computes the dominance of each input image pixel on the model's output decision. Mathematically, for a given image x with input perturbation ϵ , the Taylor series expansion of the function $f(x + \epsilon; \theta)$ is defined as:

$$f(x + \epsilon; \theta) = f(x; \theta) + \epsilon \frac{df(x; \theta)}{dx} + \mathcal{O}(\epsilon^2) \quad (7.2)$$

The higher-order terms can be neglected for very small perturbations. Hence, the above equation is updated as:

$$f(x + \epsilon; \theta) \approx f(x; \theta) + \epsilon \frac{df(x; \theta)}{dx} \quad (7.3)$$

where, term $\frac{df(x; \theta)}{dx}$ represents the input-output Jacobian matrix. Since we are computing Jacobians with respect to the true class only, we term the output matrix as Dominance matrix denoted by $D_1(x) = \frac{df(x; \theta)}{dx}$. Large values (both +ve and -ve) in the dominance matrix represent higher dominance of the corresponding image pixels in the input image x . In other words, the model's decision is highly dependent on the input image pixels with high dominance values in the dominance matrix.

Once we obtain the dominance matrix, the next step is to identify the image pixels with higher dominance values. Let p be the percentage of the most dominant image pixels. The combination of these identified pixels represents the first identified input feature $F_1(x)$. To obtain $F_1(x)$, a mask $M_1(x)$ is created using the following function:

$$M_1(x) = \begin{cases} 1 & \text{if } |D_1(x)| \geq t \\ 0 & \text{otherwise} \end{cases} \quad (7.4)$$

Here, t is the threshold obtained by sorting the dominance values of the dominance matrix for top p percentage of pixels. Next, we compute the element-wise multiplication of the mask with the input image x for obtaining feature $F_1(x)$. Mathematically, it is written as:

$$F_1(x) = M_1(x) \odot x \quad (7.5)$$

After obtaining the feature $F_1(x)$, the next step is to compute the dominance feature map $D_{m_1}(x)$ corresponding to the identified feature $F_1(x)$. These maps are used during unified model training. Here, dominance feature map $D_{m_1}(x)$ represents the dominance values of identified feature $F_1(x)$ i.e., the combination of identified dominant pixels. The dominance map $D_{m_1}(x)$ is obtained by:

$$D_{m_1}(x) = M_1(x) \odot D_1(x) \quad (7.6)$$

Next, we identify other diverse inconspicuous input features. Conceptually, diversity can be achieved by identifying features that are completely different from one another. Thus, we enforced identified features to be disjoint, i.e., $F_i(x) \cap F_j(x) = \phi$, for $i \neq j$. For this purpose, we have suppressed the identified input feature $F_1(x)$ in the input image x . This is done by setting the input image pixels to zero corresponding to the non-zero dominance values in the dominance feature map $D_{m_1}(x)$. Mathematically, it is written as:

$$x_{s_1} = \begin{cases} x & \text{if } |D_{m_1}(x)| = 0 \\ 0 & \text{if } |D_{m_1}(x)| \neq 0 \end{cases} \quad (7.7)$$

where, x_{s_1} is the output image with suppressed feature $F_1(x)$. The above-mentioned process is applied to all the training images in the dataset $\mathbf{X}$. This results in a new dataset $\mathbf{X}_{s_1}$ with suppressed features obtained corresponding to all the images. It is important to note that suppressing the identified features corresponding to all the images

in the dataset $\mathbf{X}$ will enforce the model to learn other features.

To identify other input features, we use the feature-suppressed dataset $\mathbf{X}_{s_1}$. For this purpose, we have trained a separate model using the loss function mentioned in Equation 7.1. Then, the process from Equations 7.2 to 7.7 is repeated using $\mathbf{X}_{s_1}$ in place of $\mathbf{X}$ to obtain $\mathbf{X}_{s_2}$. This process is repeated to identify input features until the stopping criteria is achieved. Figure 7.3 shows samples of the original and feature-suppressed datasets corresponding to MNIST dataset. The details of the stopping criteria are discussed in Subsection 7.4.4.

7.3.2 Diverse Unified Model Learning

Once the input features are identified, the next objective is to train a unified model that learns all the identified input features. For this purpose, we have used the dominance feature maps corresponding to all the identified features. Let k be the number of input features identified for each image x . This means we have k number of dominance feature maps i.e., $D_{m_1}(x), D_{m_2}(x), \dots, D_{m_k}(x)$. To provide supervision during training, all the dominance feature maps are combined into a single unified map by adding them together. Mathematically, it is written as:

$$D_m(x) = D_{m_1}(x) + D_{m_2}(x) + \dots + D_{m_k}(x) \quad (7.8)$$

where, $D_m(x)$ is the unified dominance feature map. Let $f_u(\cdot; \theta)$ be the unified model with parameters θ to be trained on the dataset $\mathbf{X}$ with original images. In order to enforce the model to learn all the identified input features, the following objective function is used.

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathbf{X}} [-y^T \log f_u(x; \theta) + L_D(x)] \quad (7.9)$$

Here, the first term represents the standard cross-entropy loss while the second term $L_D(x)$ represents the dominance loss. The dominance loss enforces the model to focus on the identified input features. Let $D_u(x)$ be the Dominance matrix with dominance values corresponding to the unified model $f_u(x; \theta)$. Then, in order to compute the dominance loss, the squared Euclidean distance between the unified dominance feature map $D_m(x)$ and the dominance matrix $D_u(x)$ is minimized. Mathematically, the loss

$L_D(x)$ is written as:

$$L_D(x) = ||D_m(x) - D_u(x)||_2^2 \quad (7.10)$$

Therefore, the final objective function is written as :

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathbf{X}} [-y^T \log f_u(x; \theta) + ||D_m(x) - D_u(x)||_2^2] \quad (7.11)$$

Since the unified model learns inconspicuous and diverse input features, thereby, it not only alleviates the effect of Abridged Learning but also enables the model to generalize over out-of-distribution samples.

7.4 Experimental Setup

The primary objective of this research is to remove the *shortcuts* learned by the model via learning inconspicuous and diverse input features. Our hypothesis is based on the observation that existing algorithms learn dominant features while ignoring other relevant features from the dataset within a distribution. The performance of the models suffers when dominant features are distorted/suppressed. This is because the inductive bias of the model is based on the dominant feature only. To address this, the proposed DIVINE algorithm is designed to learn the dominant features along with inconspicuous features reducing the dependence on the dominant feature only. Since the proposed learning process introduces the model to suppressed features as well, making the training diverse in nature. This ensures the generalized inductive bias of the final trained model. In order to validate this hypothesis, the experiments are performed for Abridged Learning on the feature-suppressed datasets corresponding to MNIST (LeCun *et al.*, 1998), CIFAR10 (Krizhevsky *et al.*, 2009), and TinyImageNet (Le and Yang, 2015) datasets. These feature-suppressed datasets are obtained by suppressing the identified input features (described in Section 7.3.1).

To further analyze the applicability of the proposed algorithm in real-world scenarios, we perform the experiment by evaluating the unified model on out-of-distribution samples. These samples are taken from the corrupted datasets CIFAR10-C, CIFAR100-C, TinyImageNet-C and perturbed datasets CIFAR10-P, CIFAR100-P, and TinyImageNet-P. The corrupted datasets contain 15 different corruptions corresponding to the CI-

FAR10, CIFAR100, and TinyImageNet datasets, respectively. We discuss the details of the corruption and perturbed datasets employed for evaluation below:

- **MNIST (LeCun *et al.*, 1998)** consists of 60,000 training images and 10,000 testing images, of handwritten digits from 10 different classes (0-9). Each image is a grayscale image of 28×28 resolution. The standard pre-defined protocol is used for evaluation.
- **CIFAR10 (Krizhevsky *et al.*, 2009)** contains 60,000 32×32 color images of 10 different classes with 50,000 images in training set and 10,000 images in testing set. The standard pre-defined protocol is used for evaluation.
- **TinyImageNet (Le and Yang, 2015)** is a large-scale dataset consisting of 100,000 images of 200 classes (500 for each class). The dataset consists of color images of 64×64 resolution. The standard pre-defined protocol is used for evaluation.
- **CIFAR10-C (Hendrycks and Dietterich, 2019a)** contains 15 corruptions on the CIFAR10 dataset - defocus blur, contrast, pixelate, snow, fog, glass blur, brightness, elastic transform, frost, jpeg compression, shot noise, impulse noise, zoom blur, Gaussian noise, and motion blur. The images are corrupted at five different levels with increasing severity.
- **CIFAR100-C (Hendrycks and Dietterich, 2019a)** contains 15 corruptions on the CIFAR100 dataset. The corruptions used on the CIFAR10-C dataset are also used on this dataset. The images are corrupted at five different levels with increasing severity.
- **TinyImageNet-C (Hendrycks and Dietterich, 2019b)** contains 15 corruptions on the TinyImageNet dataset. The corruptions used in CIFAR10 and CIFAR100 datasets and also used in this dataset. The images are corrupted at five different levels with increasing severity. This results in a total of 75 distinct corruptions.
- **CIFAR10-P, CIFAR100-P, and TinyImageNet-P (Hendrycks *et al.*, 2019):** The perturbed datasets CIFAR10-P, CIFAR100-P, and TinyImageNet-P modify the original CIFAR and ImageNet datasets. These datasets have smaller perturbations compared to corruption datasets and are used to measure the model's prediction stability. Each example in these datasets is a video and we measure the model's prediction consistency. Ideally, the model should not change the prediction with the increase in perturbation intensity.

7.4.1 Comparison with Existing Approaches

Since the proposed method involves the suppression of different input pixels under the supervision of dominance maps, we have performed random suppression of input features for comparison. In the random suppression (RS) method, we randomly drop $p\%$ pixels during training. Further, we have compared our method with Jacobian Regularization. In literature, Jacobian Regularization (Chan *et al.*, 2020) is primarily used for

reducing the sensitivity of input image pixels to enhance the robustness of the models. We have used this regularization method with the standard cross-entropy loss function.

Why we are not comparing DIVINE with existing Data Augmentation methods on corrupted datasets?: In the literature, various data augmentation based techniques (Hendrycks *et al.*, 2019; Yun *et al.*, 2019; Zhang *et al.*, 2018) have been presented to make the model robust on OOD samples. The results are presented on the corrupted datasets CIFAR10-C, CIFAR100-C, and TinyImageNet-C. Even though the DIVINE algorithm is evaluated on these corrupted datasets, however, it would be unfair to make comparison of the DIVINE with existing data augmentation based techniques due to the following reasons:

1. DIVINE identifies and suppresses dominant features iteratively during training, actively encouraging the model to learn inconspicuous features. It uses dominance matrices and a structured learning approach, fundamentally altering the model’s training dynamics. However, data augmentation modifies the dataset to introduce diversity without addressing the model’s inherent reliance on dominant features or its inductive biases.
2. DIVINE directly addresses the learning process to correct biases, ensuring models do not overly rely on dominant features. On the other hand, data augmentation assumes that more diverse data leads to better generalization. This approach does not explicitly consider the model’s inherent biases toward dominant features.
3. Comparing DIVINE to data augmentation techniques can mislead the interpretation of results. It might imply that DIVINE competes with augmentation methods, while in reality, the two approaches serve complementary purposes. Data augmentation techniques can potentially be combined with DIVINE to further enhance robustness and generalization, rather than being viewed as alternatives.

7.4.2 Evaluation Metrics

We compute the classification accuracy to evaluate model performance on the MNIST and CIFAR10 datasets. To evaluate the performance on the corrupted images, we report the Relative Corruption Error (Relative CE) and Relative Mean Corruption Error (Relative mCE) (Hendrycks and Dietterich, 2018).

$$\text{Relative } CE_c^f = \frac{\sum_{s=1}^5 E_{s,c}^f - E_{original}^f}{\sum_{s=1}^5 E_{s,c}^b - E_{original}^b} \quad (7.12)$$

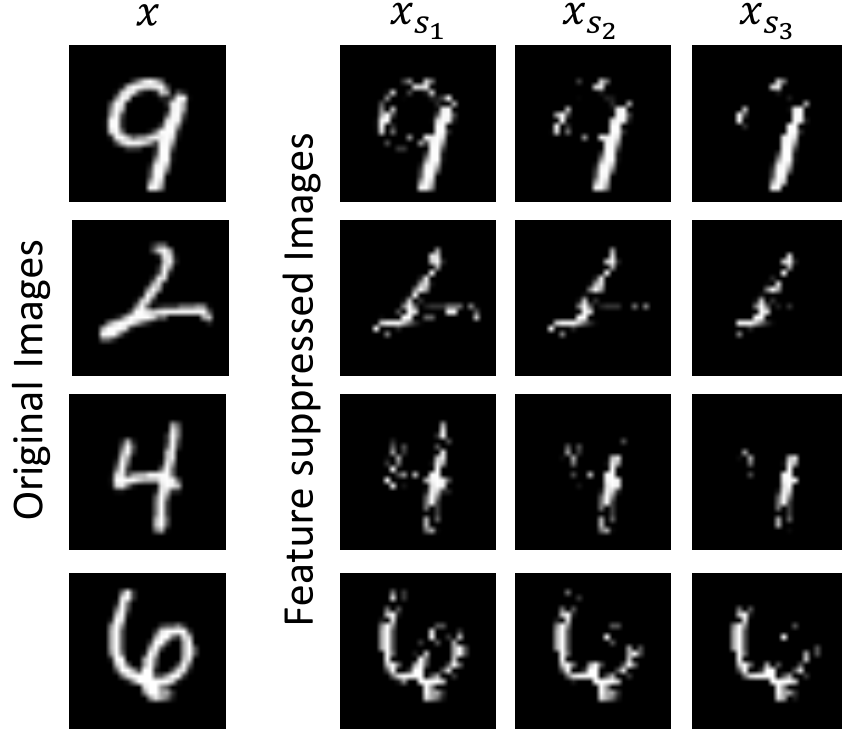


Figure 7.3: Sample original images and the corresponding feature-suppressed images on the MNIST dataset obtained using the proposed method.

where, f denotes the model to be evaluated, b denotes the baseline model obtained using Abridge Learning (AL), E_{clean}^f and E_{clean}^b denote the error obtained corresponding to the model to be evaluated and the AL model on original images. $E_{s,c}^f$ and $E_{s,c}^b$ denote error rates on corruption c at severity level s corresponding to the model to be evaluated and the AL model, respectively. A lower Relative CE indicates a higher performance over the baseline.

To evaluate the performance on the perturbed datasets, we measure the probability that two consecutive frames with different intensity of perturbations, have “flipped” or mismatched predictions. This is termed as mean Flip Rate (mFR) Hendrycks *et al.* (2019).

7.4.3 Implementation Details

For the MNIST dataset, a Convolutional Neural Network (CNN) architecture is used. The network consists of 5 convolutional layers, each followed by ReLU activation and maxpool. Two fully connected layers of dimensions 512 and 64 are added after the convolutional block followed by softmax. ResNet50, XceptionNet, and DenseNet121

Table 7.1: Classification accuracy (%) of existing and the DIVINE algorithm on the original and feature-suppressed datasets.

	Abridge Learning	Jacobian Regularization	Random Suppression	DIVINE
	MNIST			
Original	99.21	85.80	98.67	97.31
$\mathbf{X}_{s_1}$	66.53	76.52	85.56	91.41
$\mathbf{X}_{s_2}$	50.98	70.18	68.10	84.98
$\mathbf{X}_{s_3}$	40.87	62.97	54.24	75.20
Average	64.40	73.87	71.64	87.22
	CIFAR10			
Original	82.38	81.63	81.73	80.34
$\mathbf{X}_{s_1}$	64.16	65.90	63.97	67.41
$\mathbf{X}_{s_2}$	52.63	54.88	52.01	58.18
$\mathbf{X}_{s_3}$	44.45	46.87	45.19	51.83
Average	60.98	62.32	60.72	64.44
	CIFAR100			
Original	74.63	75.95	75.46	74.57
$\mathbf{X}_{s_1}$	37.44	48.31	45.21	56.31
$\mathbf{X}_{s_2}$	27.84	39.61	36.86	49.79
$\mathbf{X}_{s_3}$	15.22	26.28	14.71	33.49
Average	39.23	44.64	43.06	53.54
	Tiny-ImageNet			
Original	60.59	49.06	51.66	54.59
$\mathbf{X}_{s_1}$	40.46	39.29	43.71	43.42
$\mathbf{X}_{s_2}$	27.35	31.91	33.92	32.67
$\mathbf{X}_{s_3}$	21.67	27.28	27.94	26.66
Average	37.51	36.88	39.30	39.33

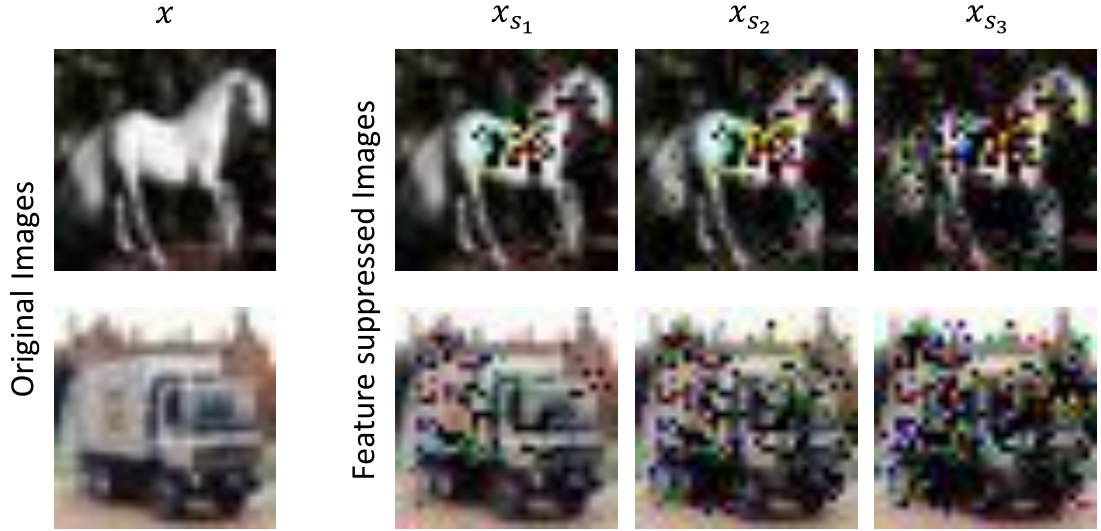


Figure 7.4: Sample original images and the corresponding feature-suppressed images on the CIFAR10 dataset obtained using the proposed method.

architectures are used for the CIFAR10, CIFAR100, and TinyImageNet datasets. Models are trained on the original and feature-suppressed datasets with a learning rate of 0.0001. For the CIFAR10 and CIFAR100 datasets, models are trained using Adam optimizer with a batch size of 32 whereas, models are trained using RMSProp optimizer with a batch size of 64 for the TinyImageNet dataset. For the MNIST dataset, the models are trained for 10 epochs and the features are suppressed with $p = 3\%$. The models on the CIFAR10 and CIFAR100 datasets are trained for 20 epochs and $p = 3\%$ is used.

For both datasets, the architectures used to train the AL models are used for training the unified model. The unified models are trained using Adam optimizer with a learning rate of 0.0001 for 20 epochs and batch size of 32. Code is implemented in Tensorflow 2.3.1. The model trainings are performed on a DGX station with Intel Xeon CPU, 256 GB RAM, and four 32 GB Nvidia V100 GPU cards. The source code is available at <https://anonymous.4open.science/r/ID-Learning-7633>.

7.4.4 Selection Criteria for Number of Feature Maps

In order to decide the number of features, we have computed the running average of the classification accuracy obtained on the original and feature suppressed dataset using AL method. We decided to iterate computing feature maps at most 3 times given the average running classification should not be below 50 percent of the classification accuracy obtained on the original dataset.

7.4.5 Implementation Details

For the MNIST dataset, a Convolutional Neural Network (CNN) architecture is used. The network consists of 5 convolutional layers, each followed by ReLU activation and maxpool. Two fully connected layers of dimensions 512 and 64 are added after the convolutional block followed by softmax. ResNet50, XceptionNet, and DenseNet121 architectures are used for the CIFAR10, CIFAR100, and TinyImageNet datasets. Models are trained on the original and feature-suppressed datasets with a learning rate of 0.0001. For the CIFAR10 and CIFAR100 datasets, models are trained using Adam optimizer with a batch size of 32 whereas, models are trained using RMSProp optimizer with a batch size of 64 for the TinyImageNet dataset. For the MNIST dataset, the models are trained for 10 epochs and the features are suppressed with $p = 3\%$. The models on the CIFAR10 and CIFAR100 datasets are trained for 20 epochs and $p = 3\%$ is used.

For both datasets, the architectures used to train the AL models are used for training the unified model. The unified models are trained using Adam optimizer with a learning rate of 0.0001 for 20 epochs and batch size of 32. Code is implemented in Tensorflow 2.3.1. The model trainings are performed on a DGX station with Intel Xeon CPU, 256 GB RAM, and four 32 GB Nvidia V100 GPU cards.

7.5 Results and Analysis

The proposed DIVINE algorithm is evaluated on two types of datasets: (i) feature-suppressed datasets and (ii) corruptions. In the first set of evaluations, we validate our assertion of “Abridge Learning” using the MNIST, CIFAR10, CIFAR100, and TinyImageNet datasets. For corruptions, we showcase results on CIFAR10-C, CIFAR100-C, and TinyImageNet-C datasets. We further compare the DIVINE algorithm with an algorithm proposed by Carter *et al.* (2021).

7.5.1 Evaluation on Feature-suppressed Datasets

This experiment is performed to validate that a model trained using conventional methods relies on the dominant features (which are easy to learn) during classification,

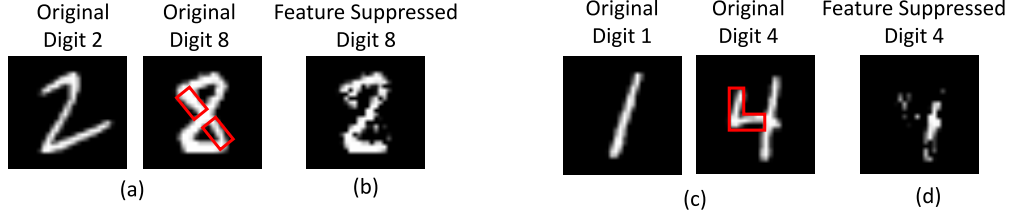


Figure 7.5: Visualizations of the semantically relevant features learned by the model. (a) shows the strokes learned by the model, which distinguishes digit 8 from 2 and (b) shows the feature-suppressed image. Similarly (c) and (d) show the distinguishing strokes learned by the model and the feature-suppressed image.

thereby reducing the performance of the model when these dominant input features are missing. The performance of the AL model trained on original images is evaluated on the testing set of original and feature-suppressed datasets, i.e., $\mathbf{X}$, $\mathbf{X}_{s_1}$, $\mathbf{X}_{s_2}$ and $\mathbf{X}_{s_3}$. Dataset $\mathbf{X}_{s_1}$ has images with one suppressed dominant input feature in each image. Similarly, datasets $\mathbf{X}_{s_2}$ and $\mathbf{X}_{s_3}$ have images with two and three suppressed dominant input features in each image, respectively. Table 7.1 shows the performance of the AL models corresponding to the MNIST, CIFAR10, CIFAR100, and TinyImageNet datasets. It is observed that the performance of the AL models degrades significantly on the feature-suppressed datasets. For instance, the performance of the AL model trained on the MNIST dataset drops from 99.21% to 66.53% on feature-suppressed dataset $\mathbf{X}_{s_1}$ (32.68% drop), which further degrades to 50.98% on feature-suppressed dataset $\mathbf{X}_{s_2}$. This shows that the performance of the models trained using conventional methods is heavily dependent on the dominant input features. On the other hand, the performance of the unified model drops by only 5.90% when evaluated on $\mathbf{X}_{s_1}$. As seen in Table 7.1, the unified model trained using the proposed DIVINE algorithm performs well on the feature-suppressed datasets.

Results of the unified model in Table 7.1 are compared with random suppression, and the Jacobian regularization method (Chan *et al.*, 2020). Both approaches are used to enhance the robustness of the models. It is observed that existing approaches perform better than the AL model, especially on the MNIST dataset. However, the performance is not as good as that obtained using the unified model. In the random suppression method, there is no supervision to the model for learning a diverse set of features. While in the Jacobian regularization method, the model reduces its dependency on the dominant features during training, it is not able to learn the inconspicuous features.

Table 7.2: Comparison of relative corruption error obtained using existing and proposed methods for each corruption type on the CIFAR10-C and CIFAR100-C dataset.

Corruptions	CIFAR10-C			CIFAR100-C		
	Jacobian Regularization	Random Suppression	DIVINE	Jacobian Regularization	Random Suppression	DIVINE
Defocus Blur	102.35	108.77	92.90	105.88	108.38	102.08
Contrast	100.19	103.55	99.64	103.86	114.32	122.66
Pixelate	95.09	113.08	75.58	108.59	106.95	74.61
Snow	91.57	81.15	81.38	97.17	99.32	78.91
Fog	105.04	111.62	99.17	111.01	110.61	110.52
Glass Blur	96.11	87.62	73.69	98.51	98.49	82.99
Brightness	106.40	133.70	101.00	129.75	136.48	125.60
Elastic	103.84	109.48	97.95	107.35	106.55	96.89
Frost	100.47	113.05	88.12	104.53	111.80	83.74
JPEG	86.40	93.95	70.86	93.23	91.82	73.24
Shot Noise	95.65	111.77	72.62	101.47	107.27	76.46
Impulse Noise	91.70	92.17	73.02	105.16	96.99	78.15
Zoom Blur	99.16	111.06	95.85	97.47	98.42	89.50
Gaussian Noise	95.49	110.70	73.32	101.74	107.97	79.18
Motion Blur	105.37	116.31	99.40	107.86	113.26	98.38
Relative mCE	98.32	106.29	86.30	104.90	107.24	91.53

Figure 7.5 (a) & (b), show that the discriminative stroke of digit ‘8’ (highlighted in red) is the most dominant feature differentiating it from digit ‘2’, and is therefore suppressed. Similar observations can be made for digits ‘4’ and ‘1’ in Figure 7.5 ((c) & (d)). Figure 7.4 effectively illustrates the core principle of the DIVINE algorithm: iterative feature suppression forces the model to learn inconspicuous, diverse features that enhance generalization and robustness. The key insights from the Figure 7.4 are as follows:

1. The original images contain all features, including dominant and inconspicuous ones, which the model initially uses for classification.
2. Feature-suppressed images represent the results of successive iterations of feature suppression. Considering the first row representing the sample of a class horse:
 - (a) The image x_{s_1} represents that the model focuses on the body of the horse (dominant feature), and this is removed in x_{s_1} .
 - (b) After suppressing the first dominant feature, the model identifies and suppresses the second dominant feature i.e., parts of the horse’s legs, as shown in x_{s_2} .
 - (c) In the third iteration, the next set of dominant features is suppressed, which are other structural elements like part of the tail or mane, as shown in x_{s_3} .

3. The features learned by the model during suppression are semantically meaningful. For example: The model’s attention transitions from the horse’s body and legs (dominant features) to the tail, mane, and even background details (inconspicuous features). This process ensures that the model does not overly depend on any single set of dominant features for classification.

The visualization of the dominance matrix computed corresponding to both the AL and unified models is shown in Figure 7.6. We have made the following observations from Figure 7.6:

1. The dominance matrices $D_1(x)$ corresponding to AL models are highly concentrated on specific dominant input features. This confirms that AL models primarily rely on dominant features for making predictions, neglecting other potentially useful inconspicuous features.
2. The matrices $D_2(x)$ and $D_3(x)$, which correspond to models trained on feature-suppressed datasets, show a shift in focus from dominant features to inconspicuous features.
3. The unified dominance matrix $D_u(x)$, representing the fully trained DIVINE model, shows a balanced focus across all identified input features. This includes both dominant and inconspicuous features, ensuring robustness and generalization.
4. The broader focus in $D_u(x)$ aligns with the improved performance of DIVINE on feature-suppressed datasets (as highlighted in Table 7.1) and on out-of-distribution datasets.

Overall, Figure 7.6 provides a clear visualization of the progression from narrow feature focus (AL) to comprehensive feature learning (DIVINE). The shift in dominance matrices underscores the success of DIVINE in mitigating Abridge Learning by promoting the learning of diverse and inconspicuous features. This balanced learning enhances robustness and generalization, as demonstrated in the experimental results.

Ablation Study for parameter p : On updating the values of p from 3% to 10%, the performance of the unified model degrades on feature-suppressed datasets. Since, the model prediction is dependent only on a few input pixels, setting a higher value of p results in suppressing of dominant as well as other input features, which in turn decreases the performance of the unified model on the feature-suppressed datasets.

Table 7.3: Classification accuracy (%) obtained using Jacobian Regularization (JR), Random Suppression (RS), and DIVINE algorithm on the TinyImageNet-C dataset for different corruptions.

Corruptions	Jacobian Regularization	Random Supression	DIVINE
Defocus Blur	21.07	19.97	29.43
Contrast	12.37	12.51	18.70
Pixelate	27.98	31.79	39.53
Snow	25.16	26.83	32.8
Fog	21.46	23.37	33.09
Glass Blur	28.33	30.36	32.14
Brightness	27.02	28.33	36.14
Elastic	21.23	21.47	30.89
JPEG	26.30	29.07	37.25
Shot Noise	35.77	37.68	43.09
Impluse Noise	29.71	34.09	39.14
Zoom Blur	20.07	18.83	28.20
Gaussian Noise	34.75	36.99	41.84
Motion Blur	21.47	22.11	30.87
Mean	25.19	26.67	33.79

7.5.2 Evaluation on Corruptions and Perturbations

This experiment is performed to evaluate the generalizability and robustness of the proposed unified model on out-of-distribution images. The performance on corruptions and perturbations are discussed below:

Performance on Corruption We have reported the relative corruption error (Relative CE) and the relative mean corruption error (Relative mCE) on the CIFAR10-C and CIFAR100-C datasets. The results are shown in Table 7.2. The proposed unified model outperforms random suppression and Jacobian Regularization methods on the corruption datasets. The proposed unified model gives a Relative mCE of **86.30** and **91.53** corresponding to the CIFAR10-C and CIFAR100-C datasets, respectively. On comparing the performance of CIFAR10-C and CIFAR100-C datasets for individual corruptions, the proposed unified model trained with the DIVINE method outperforms other methods on 14 corruptions (excluding snow corruption on the CIFAR10-C and contrast corruption on the CIFAR100-C) and gives a comparable performance on the

Table 7.4: Classification accuracy (%) obtained using Abridged Learning and DIVINE algorithm on the CIFAR10-P and CIFAR100-P datasets for different perturbations.

mFR %	CIFAR10		CIFAR100	
	Abridged Learning	DIVINE	Abridged Learning	DIVINE
Brightness	1.33	1.22	2.98	1.03
Gaussian Noise	5.21	3.86	15.3	2.21
Motion Blur	11.26	9.9	14.74	2.1
Rotate	8.29	6.24	11.12	3.09
Scale	9.55	7.99	13.2	4.87
Shot Noise	6.4	4.82	17.96	2.9
Snow	3.75	2.96	6.59	1.15
Tilt	3.13	2.49	5.52	1.55
Translate	15.63	13.53	28.34	11.79
Zoom Blur	0.79	0.67	1.78	0.32
Overall mFR	6.534	5.368	11.753	3.101

snow and contrast corruption corresponding to CIFAR10-C dataset, respectively.

Table 7.1 shows that the performance of all the existing methods is comparable to the original images. However, random suppression and Jacobian regularization methods fail to generalize well on out-of-distribution images as shown in Table 7.2 due to the following reasons:

1. In case of random suppression, the model lacks supervision to focus on meaningful inconspicuous features. This approach does not ensure that suppressed features are either dominant or relevant, potentially leading to suboptimal learning.
2. In case of Jacobian Regularization (JR), it effectively reduces dependency on dominant features, however, it does not actively guide the model to learn inconspicuous features, leaving a gap in robust feature learning.

To test our method on a large-scale dataset, we computed the performance on the TinyImageNet-C dataset and report the results obtained in Table 7.3. On the TinyImageNet-C dataset, DIVINE yields an absolute improvement of 25.54% and 19.02% over the random suppression and Jacobian regularization methods, respectively as shown in Table 7.3. We observe the robustness of the proposed DIVINE algorithm against a variety of corruptions. These results are illustrated for three different learning methods namely- Jacobian Regularization, Random Suppression, and the proposed DIVINE algorithm.

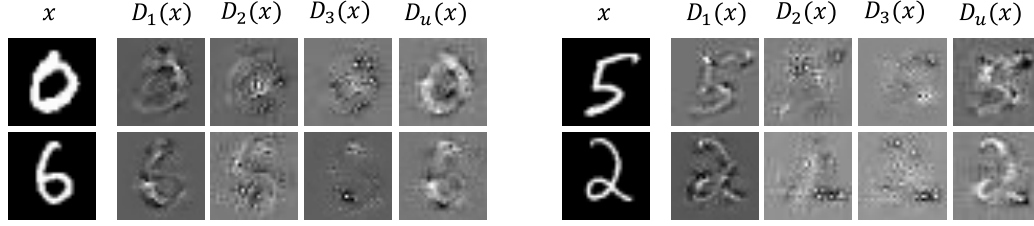


Figure 7.6: Illustration of dominant features and inconspicuous features obtained in the MNIST dataset. Sample original images x and the corresponding dominance matrices $D_1(x)$, $D_2(x)$, $D_3(x)$, and $D_u(x)$. $D_1(x)$ and $D_u(x)$ are obtained corresponding to the AL and unified model trained on the original dataset. In contrast, $D_2(x)$ and $D_3(x)$ are obtained corresponding to the model trained on feature-suppressed datasets.

From the table, it is clearly visible that the proposed algorithm outperforms other algorithms on all corruptions. We also achieve significantly better mean classification accuracy using the proposed DIVINE algorithm. This clearly describes the applicability of DIVINE algorithm on large-scale datasets as well.

Performance on Perturbations We have reported the Flip Rate (FR) and the Overall mean flip rate (Overall mFR) on the CIFAR10-P, CIFAR100-P, and TinyImageNet-P datasets. The results are shown in Table 7.4 and 7.5. The proposed unified model outperforms abridge learning on the perturbed datasets. The proposed unified model gives an overall mFR of **5.36%**, **3.10%**, and **21.85%** corresponding to the CIFAR10-P, CIFAR100-P, and TinyImageNet-P datasets, respectively. On comparing the performance of individual perturbations on all datasets, the proposed unified model trained with the DIVINE method outperforms abridge learning. We have also compared the performance of DIVINE with existing methods and the results are shown in the Table 7.6. The proposed algorithm DIVINE outperforms the second best algorithm IpMix Huang *et al.* (2023) by **mFR 1.2%** and achieves the **state-of-the-art (sota)** results.

The visualization of the dominance matrix obtained corresponding to the unified model and existing approaches on the corrupted images of the CIFAR10 dataset are shown in Figure 7.7. We can see that the unified model focuses on multiple input features/regions while the existing approaches fail to focus diversely. On observing the relative corruption error corresponding to impulse noise corruption in Table 7.2, it is found that the error for the proposed method is 73.02, which is 18.68 and 19.15 less than Jacobian regularization and random suppression methods, respectively. This shows

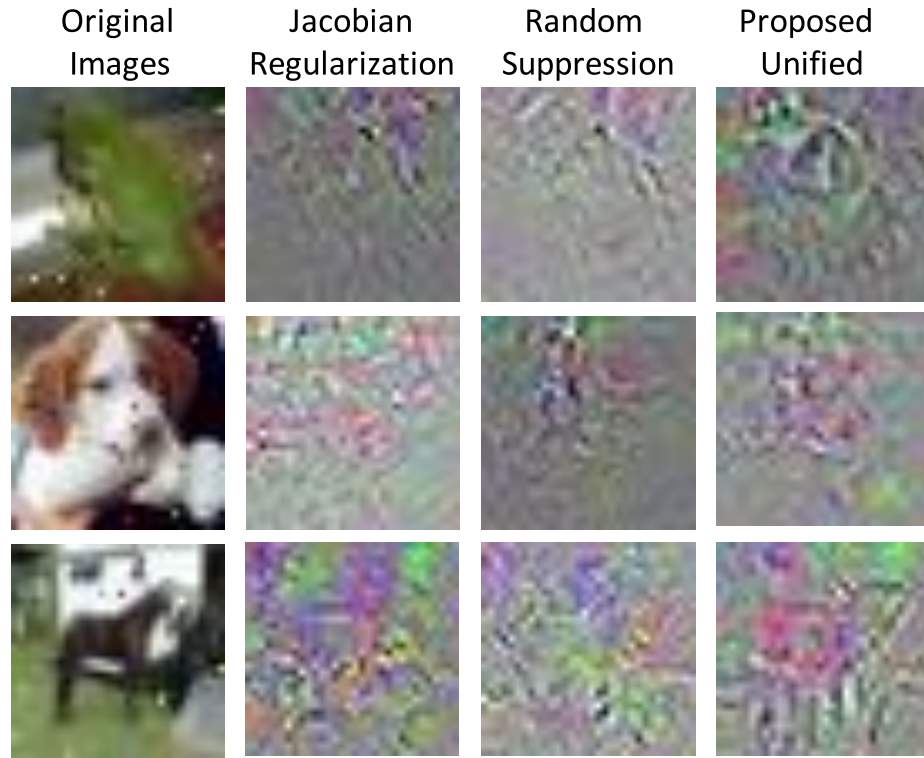


Figure 7.7: Sample images corrupted with impulse noise and the corresponding dominance matrices obtained using Jacobian regularization, random suppression, and the proposed unified model.

the applicability of the proposed method in real-world scenarios where external corruptions are common. Additionally, the proposed method can be used in combination with existing approaches for improving robustness.

Comparison with Carter et al. (NeurIPS 2021) (Carter *et al.*, 2021): Carter et al. Carter *et al.* (2021) have shown that only 5% spurious pixel subsets in an image are enough for confident prediction. These pixel subsets may be meaningless to humans and lead to over-interpretation by the model. This validates our assumption of abridge

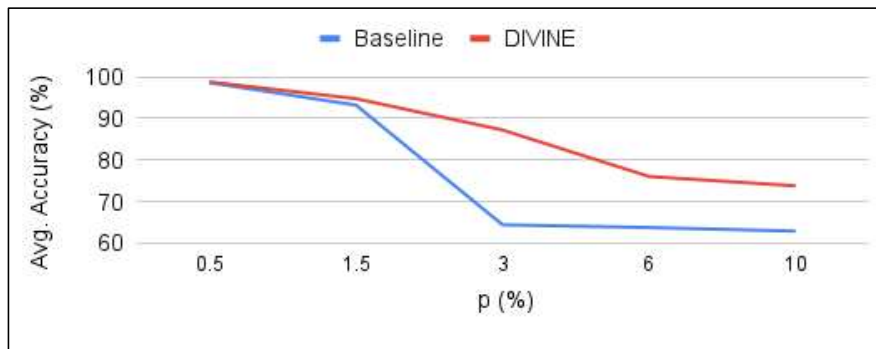


Figure 7.8: Plot representing the trend line of the average classification accuracy corresponding to baseline and proposed algorithm on different values of p on the MNIST dataset.

Table 7.5: Classification accuracy (%) obtained using Abridged Learning and DIVINE algorithm on the TinyImageNet-P dataset for different perturbations.

mFR %	TinyImageNet	
	Abridged Learning	Unified Model
Brightness	12.09	9.53
Gaussian Noise	28.56	18.58
Gaussian Noise V3	56.82	42.36
Rotate	45.49	30.19
Shear	37.83	24.19
Shot Noise V2	41.11	26.99
Snow	15.66	11.26
Speckle Noise	26.4	15.26
Speckle Noise V3	53.97	38.08
Translate	41.92	31.51
Gaussian Blur	5.17	3.91
Gaussian Noise V2	47.06	32.98
Motion blur	7.66	5.36
Scale	40.3	29.87
Shot Noise	32.09	20.14
Shot Noise V3	50.29	34.85
Spatter	17.5	12.22
Speckle Noise V2	44.1	28.65
Tilt	25.26	15.45
Zoom Blur	8.73	5.74
Overall mFR	31.9005	21.856

learning during training of the model. The authors further used an ensemble method to mitigate the problem of abridge learning. In order to showcase the effectiveness of the proposed DIVINE algorithm, we have also compared its performance on CIFAR10-C dataset. The DIVINE algorithm achieves a relative mCE of 86.30 which outperforms the ensemble method (Carter *et al.*, 2021) by a margin of 8.62. In general, ensemble methods reduce overfitting and improve model performance. However, they do not necessarily make the model robust towards out-of-distribution samples. By enforcing the learning of inconspicuous input features, the DIVINE algorithm offers better robustness.

Comparison with LLE Li *et al.* (2023c): In the literature, Li *et al.* (2023c) proposed a Last Layer Ensemble (LLE) method to mitigate the multiple shortcuts present

Table 7.6: Comparison of Mean Flip Rate (mFR) of DIVINE with existing algorithms on the CIFAR100-P dataset.

Method	mFR %
MixUp Zhang <i>et al.</i> (2018)	8.9
CutOut DeVries (2017)	11.9
CutMix Yun <i>et al.</i> (2019)	12.0
AugMix Hendrycks <i>et al.</i> (2019)	5.6
PixMix Hendrycks <i>et al.</i> (2022)	5.6
IpMix Huang <i>et al.</i> (2023)	4.3
DIVINE	3.1

in the dataset, which is also one of the objectives of the proposed DIVINE algorithm. However, there are several fundamental differences between DIVINE and LLE Li *et al.* (2023c), which makes DIVINE better. These differences are as follows:

1. DIVINE effectively balances the learning of dominant and inconspicuous features, ensuring generalization even under severe perturbations. On the other hand, LLE focuses on mitigating shortcut reliance, which does not explicitly guarantee balanced feature learning.
2. LLE is primarily evaluated on two new datasets, UrbanCars with controlled spurious cues like background and co-occurring objects) and ImageNet-W (an ImageNet variant with watermarked images) and included a wide range of baseline methods. On the other hand, DIVINE has been evaluated on a more diverse set of datasets, including MNIST, CIFAR10, CIFAR100, TinyImageNet, and their corrupted and perturbed counterparts (CIFAR10-C, CIFAR10-P, CIFAR100-C, CIFAR100-P, TinyImageNet-C, and TinyImageNet-P). This broad range of datasets ensures robustness and generalizability across various real-world conditions, making our evaluation more extensive in terms of dataset variety.
3. Unlike LLE, which introduces additional classifiers and a distributional shift classifier, DIVINE provides a simpler and more efficient framework that can be easily integrated into existing training pipelines. Furthermore, our work outperforms state-of-the-art data augmentation techniques (e.g., MixUp, CutMix, AugMix, PixMix, and IpMix) in terms of mean flip rate (mFR), demonstrating its robustness without relying on data augmentation alone.
4. LLE is applicable to scenarios where the shortcuts are known as given in UrbanCars and ImageNet-W datasets. Manually identifying the shortcuts in the dataset is a tedious task and non-scalable. In real-world scenarios, the shortcuts present in the dataset and learned by the models are unknown. On the other hand, the proposed DIVINE is not dependent on the known shortcuts and is able to handle multiple unknown shortcuts.
5. While DIVINE incurs additional computational costs due to Jacobian computations, the complexity of maintaining ensemble classifiers in LLE may lead to

Table 7.7: Training time of Abridged Learning and proposed DIVINE method on CIFAR10 dataset.

	Abridged Learning	DIVINE
Time per Epoch (in seconds)	73	177
Total Training Time (in minutes)	25	60

higher memory and computational overhead.

7.5.3 Ablation Experiment to Visualize the Trend of Parameter p

We conduct a series of experiments for varying values of p , specifically at 0.5%, 1.5%, 3%, 6%, and 10%. The trend observed in average classification accuracy for these values is depicted in Figure 7.8, and detailed results are presented in Table 1. From this representation, we notice a slight decrease in accuracy at $p=0.5\%$ and $p=1.5\%$ for the baseline model, indicating that dominant features are still present in the datasets with feature suppression at these lower p values. However, at $p=3\%$, there is a noticeable decline in the baseline model’s performance, highlighting the successful elimination of dominant features in the dataset. Consequently, we have selected $p=3\%$ as the optimal value for conducting our experiments.

7.5.4 Computational Runtime

We have calculated the temporal cost associated with the MNIST, CIFAR10, CIFAR100, and TinyImageNet datasets during the first phase. In this phase, the model undergoes training on the dataset with suppressed features, and this process is repeated three times. The MNIST dataset requires 10 epochs for training, with each epoch taking approximately 14 seconds to complete. The entire training process for MNIST, therefore, takes around 420 seconds, equivalent to 7 minutes. In the case of CIFAR10 and CIFAR100 datasets, each epoch lasts about 61 seconds, while for the TinyImageNet dataset, it is around 935 seconds per epoch. These time durations present opportunities for optimization, possibly through the application of methods like the one proposed by (Wang *et al.*, 2020), which involves pruning the network at the initial stage, prior to training. It is important to note that the time required for inference (testing) remains consistent between

the proposed unified model and the AL model. Further, we compute the time complexity of Abridge Learning and the proposed DIVINE algorithm for each epoch as well as the total training time. From Table 7.7 we observe that overhead training time is 104 seconds for each epoch, which is mostly spent on the computation of Jacobians. Since, the proposed DIVINE method increase the computation time over Abridge Learning, we consider this a limitation of the proposed method the minimization of this overhead can be explored in the future work.

7.6 Limitations

We highlight the following limitations of the proposed DIVINE algorithm:

- Though the proposed method mitigates Abridged Learning, it is computationally expensive as it uses iterative suppression and retraining process to identify inconspicuous and diverse features and training of the models on the same data. However, this iterative mechanism is fundamental to DIVINE’s ability to learn both dominant and inconspicuous features, which significantly improves robustness and generalization. To mitigate this limitation in future work, we are exploring optimization strategies, such as gradient approximation techniques, selective iteration schemes, and model pruning or early stopping.
- The proposed idea is validated only for the image modality in classification setting. However, the proposed approach can be extended for more modalities and different tasks.
- The proposed method DIVINE is not applicable to those abridge learning problems where the dominant features are not semantic. For example, in ColoredMNIST dataset, the spurious correlation is due to color attribute of the digits, which is not a semantic feature and DIVINE is capable of only highlighting the semantic features in the input data. While DIVINE is effective for scenarios where dominant features are semantically relevant, addressing non-semantic biases requires a different strategy. Future work could incorporate bias mitigation frameworks, feature attribution analysis, and data augmentation methods to deal with non-semantic features.

7.7 Summary

Conventional deep learning algorithms typically prioritize enhancing classification accuracy, which can result in inadequate learning and poor generalization to out-of-distribution samples. In this research, we introduce a unique, holistic learning approach

called *Diverse and Inconspicuous Learning* (DIVINE) which focuses on maximizing learning from a given set of inputs and learning as many discriminative features as possible. We validate DIVINE’s effectiveness through extensive experiments across various datasets, including MNIST, CIFAR10, CIFAR100, TinyImageNet, and their corrupted and perturbed versions. The proposed approach consistently outperforms state-of-the-art methods in robustness and out-of-distribution generalization. The results reveal that the dominance maps generated via our method provide superior guidance for learning a rich set of input features. Consequently, our model demonstrates enhanced generalizability and robustness, particularly in the presence of out-of-distribution samples. We posit that this comprehensive style of learning ensures more reliable model predictions, especially in real-world situations where data corruption and distribution shifts can significantly impair performance.

Chapter 8

Conclusion

Artificial Intelligence (AI) has rapidly evolved into a transformative technology shaping nearly every aspect of human life. Despite its immense potential, the widespread deployment of AI systems has exposed several foundational challenges related to privacy, robustness, and trustworthiness. This thesis has undertaken a holistic exploration of these challenges, presenting an integrated research framework that bridges privacy preservation, media authenticity, model adaptability, and generalization. Collectively, the contributions of this work form a cohesive effort toward designing AI systems that are secure, interpretable, and ethically aligned with societal needs. Rather than treating each challenge in isolation, the thesis demonstrates how privacy protection, robustness, and generalization are interdependent aspects of building reliable AI and offers principled solutions that advance all these dimensions simultaneously.

This thesis has systematically addressed some of the most pressing issues, including privacy breaches, the proliferation of fake media, and the limited robustness and generalizability of AI models in real-world scenarios. Through novel contributions, we have advanced the field's understanding and proposed practical solutions to mitigate these challenges effectively. The main contributions of the dissertation are summarized below:

1. **Attribute and Identity Anonymization for Privacy Preservation:** Privacy concerns in AI-powered face analysis are increasingly significant, especially with the ease of extracting sensitive information from facial images. To address this, we proposed a privacy-preserving algorithm that leverages adversarial perturbations to selectively anonymize facial attributes and identity information while maintaining visual integrity. Unlike existing methods, our approach does not rely on reference images and offers precise control over which attributes are anonymized or retained. This contribution empowers users with greater control over their data, striking a balance between privacy and utility. Empirically, on the CelebA dataset, the suppression of sensitive attributes such as Gender, Attractive, and Smiling achieved over 96% suppression accuracy, with classifier accuracy for attributes like Attractive dropping from 89.31% to 0.28% after anonymization. For identity anonymization, the proposed algorithm achieved a true positive rate (TPR) of 0.002 on the LFW dataset using the ArcFace model, establishing a new state-of-the-art benchmark for identity suppression.

2. **Low-Quality Deepfake Detection:** The misuse of Deepfake technology has become a significant threat, particularly with the dissemination of manipulated content on social media platforms that apply compression. Existing Deepfake detection algorithms often fail on low-quality, compressed media. To address this, we proposed an algorithm that enhances previously unseen and imperceptible artifacts, learning from subtle features often lost during compression. By incorporating a visibility matrix, our method enhances detection capabilities, effectively tackling the challenges posed by low-quality Deepfakes and ensuring media authenticity. On the FaceForensics++ dataset under conditions of strong compression, the proposed algorithm, when trained on high-quality videos and evaluated on low-quality counterparts, demonstrated a 17.34% increase in classification accuracy relative to the approach of Kumar *et al.* (2020).
3. **Data Fine-Tuning:** In scenarios where commercial off-the-shelf (COTS) systems act as black boxes, fine-tuning model parameters is often infeasible due to their inaccessibility. To overcome this, we introduced a data fine-tuning technique that adapts input data to align with the model's inherent behavior, enhancing performance without modifying the model itself. Experiments on CelebA and LFW demonstrated accuracy improvements of 17%–29% improvements across all tested attributes in the inter-dataset setting, showing its effectiveness at adapting black-box models to new datasets without modifying the model itself. Furthermore, our Feature-Guided Perturbation (FGP) method addresses the issue of data shift by refining the feature space, ensuring reliable performance across diverse scenarios. These techniques provide a scalable and efficient solution for improving the adaptability of black-box systems in real-world applications.
4. **Diverse and Inconspicuous Learning (DIVINE):** Traditional deep learning algorithms often overfit dominant features, leading to poor generalization in out-of-distribution scenarios. To address this limitation, we introduced the DIVINE framework, which encourages models to learn a rich set of discriminative features through the use of dominance maps. Experimental results across several datasets, including perturbed and corrupted variants, demonstrated the robustness and generalization capabilities of DIVINE. This approach sets a new standard for designing deep learning models that are resilient to variations and disruptions in real-world applications. For example, on the CIFAR100-P dataset, DIVINE achieved the lowest mean Flip Rate (mFR) of 3.1% surpassing the previous best method IpMix Huang *et al.* (2023) at 4.3% on out-of-distribution test set, underscoring its improved generalization.

The contributions of this thesis have far-reaching implications for the development of ethical and reliable AI systems. By addressing privacy concerns, improving model adaptability, and enhancing the detection of manipulated media, this research contributes to the creation of AI systems that are both technically robust and socially responsible. The methodologies and frameworks proposed here can serve as a foundation for industry and academia to develop next-generation AI solutions capable of meeting the demands of diverse and dynamic environments.

Taken together, the methods proposed in this dissertation demonstrate that achieving privacy preservation, robustness against adversarial manipulations, adaptability of black-box systems, and generalization under distribution shifts are not independent challenges but deeply interlinked. By delivering measurable improvements across all four dimensions, this work establishes a comprehensive framework for building AI systems that are not only accurate but also trustworthy, resilient, and compliant with ethical and legal standards. Looking forward, the insights and frameworks presented here open new avenues for research that integrates privacy, fairness, robustness, and transparency, ensuring that AI technologies can be deployed responsibly in high-stakes, real-world environments.

8.1 Future Research Directions

The research presented in this thesis has contributed to privacy preservation, Deepfake detection, and robust model training through adversarial perturbations, data fine-tuning, and feature-guided learning. While the proposed solutions address pressing challenges, the rapid progress in Generative AI, including diffusion models and large multimodal foundation models opens new opportunities and necessitates further exploration. The following directions outline natural extensions of the work presented in the thesis:

1. **Robustness Against Adversarial Defenses and Identity Drift:** While the proposed perturbation successfully anonymizes identity in a white-box setting, its robustness against adversarial defenses, such as adversarial training or input denoising, has yet to be fully explored. Real-world applications may involve systems that employ such defenses, which could reduce the effectiveness of the anonymization. Furthermore, while the perturbation effectively removes identity information under an $\mathcal{L}_p$ -norm constraint, it may cause the anonymized embedding to drift towards feature regions of other identities, potentially leading to unintended re-identification. Future work should focus on making the anonymization process more resilient to adversarial defenses and identity drift. This can involve developing adaptive strategies for defense-aware anonymization and regularizing perturbations to prevent the drift of anonymized representations into feature regions of real identities.
2. **Robustness to Image Degradations and Transferability Across Models:** The proposed framework is designed to be robust against common image degradations such as resizing, blurring, or downsampling, which typically affect Deepfake detection methods. However, evaluating this robustness across diverse degradation settings and ensuring that perturbations remain effective in real-world conditions is an important next step. Additionally, the learned perturbation designed to enhance the performance of a black-box system is typically optimized for a specific

model and may not generalize effectively to other architectures or classifiers. Different models may have distinct feature representations and decision boundaries, limiting the generalizability of the perturbations. Future work should address both the degradation-resilience and cross-model transferability of learned perturbations, potentially by developing model-agnostic fine-tuning techniques or ensemble-based methods that improve generalization across architectures.

3. **Scalable and Generative Anonymization:** Future work can explore lightweight anonymization frameworks that integrate adversarial perturbations with diffusion or transformer based generative models. Such systems would enable fine-grained, user-controlled anonymization at the point of capture or upload, while remaining resilient to reconstruction or inversion attacks made possible by foundation models. Approaches such as gradient approximation, selective iteration, pruning strategies, and amortized optimization may further improve scalability and real-time applicability. Extending these techniques to federated and edge computing settings offers strong potential for distributed privacy guarantees.
4. **Robust Deepfake Detection in the Generative AI Era:** As Deepfake pipelines shift from GANs to diffusion-based and multimodal generative models, future detectors must be artifact-agnostic and continually adaptive. Building on the unseen artifact framework, research should develop continual learning approaches capable of evolving with generative pipelines. Synthetic datasets generated by large models may serve as controlled benchmarks to train detectors that remain robust across modalities, domains, and evolving manipulation techniques.
5. **Perturbation-Guided Foundation Model Adaptation:** Perturbation-based fine-tuning and feature-guided learning can be extended to large foundation models (e.g., CLIP, vision–language models). Rather than retraining millions of parameters, perturbations could offer rapid, interpretable, and task-specific personalization across domains such as face recognition, speech, or medical imaging. Coupling these perturbations with latent-space manipulations in diffusion or GAN models may further improve robustness, fairness, and interpretability.
6. **Generalization to Heterogeneous and Real-World Data:** Beyond controlled datasets, it is essential to test perturbation-based methods on cluttered scenes, low-resolution imagery, occluded or multi-object settings, and cross-domain tasks. Robustness against naturally occurring covariates such as pose variation, sensor noise, or adversarial shifts should be systematically studied. Extending frameworks like DIVINE to leverage non-dominant latent features and generative augmentation could enhance generalization and resilience.
7. **Cross-Modal and Multi-Task Extensions:** Perturbation-guided privacy and robustness strategies should be extended beyond facial imagery to video, speech, and medical data, where security and interpretability are equally critical. Multi-task frameworks that jointly optimize privacy preservation, robustness against adversarial shifts, and generalization hold promise for broad and unified application.
8. **Efficient and Ethical Real-World Deployment:** The translation of these methods into deployable, real-time pipelines for surveillance, mobile applications, and edge devices necessitates hardware and software co-optimization to balance computation, latency, and security. At the same time, future work should integrate

ethical and societal considerations, including explainable systems for user-in-the-loop control, privacy-preserving tools for social media, and collaboration with policymakers to establish responsible standards for generative AI.

Bibliography

1. (2018). Questions and Answers – General Data Protection Regulation. https://ec.europa.eu/commission/presscorner/detail/en/MEMO_18_387/.
2. (2019). FACE, BODY, VEHICLE, AND LICENSE PLATE NUMBER RECOGNITION. <https://ntechlab.com/>.
3. (2023). FaceCheck.ID Find People Online by Photo. <https://facecheck.id/>.
4. (2023). Steer clear of identity theft on social media. <https://www.allstateidentityprotection.com/content-hub/how-do-criminals-use-social-media-to-steal-information>.
5. (2024). Global social media stats. <https://tinyurl.com/5n7c4u3v>.
6. (2024). Identity theft. <https://www.usa.gov/identity-theft>.
7. **Abdulnabi, A. H., G. Wang, J. Lu, and K. Jia** (2015). Multi-task cnn model for attribute prediction. *IEEE TM*, **17**(11), 1949–1959.
8. **Afchar, D., V. Nozick, J. Yamagishi, and I. Echizen**, Mesonet: a compact facial video forgery detection network. In *IEEE International Workshop on Information Forensics and Security*. 2018.
9. **Agarwal, S., H. Farid, Y. Gu, M. He, K. Nagano, and H. Li**, Protecting world leaders against deep fakes. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019.
10. **Akata, Z., S. Reed, D. Walter, H. Lee, and B. Schiele**, Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2015.
11. **Alam, M. M.** (2023). Social media and financial crime - a growing threat and the online safety act 2023 (uk). <https://www.linkedin.com/pulse/social-media-financial-crime-growing-threat-online-tltde/>.
12. **Amos, B., B. Ludwiczuk, and M. Satyanarayanan** (2016). Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118.
13. **Bai, H., R. Sun, L. Hong, F. Zhou, N. Ye, H.-J. Ye, S.-H. G. Chan, and Z. Li** (2021). Decaug: Out-of-distribution generalization via decomposed feature representation and semantic augmentation. *AAAI Conference on Artificial Intelligence*.
14. **Barattin, S., C. Tzelepis, I. Patras, and N. Sebe**, Attribute-preserving face dataset anonymization via latent code optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023.

15. **Bayar, B.** and **M. C. Stamm**, A deep learning approach to universal image manipulation detection using a new convolutional layer. *In ACM Workshop on Information Hiding and Multimedia Security*. 2016.
16. **Benkert, R., M. Prabhushankar,** and **G. AlRegib**, Forgetful active learning with switch events: Efficient sampling for out-of-distribution data. *In IEEE International Conference on Image Processing*. 2022.
17. **Berg, T.** and **P. Belhumeur**, Poof: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2013.
18. **Beveridge, J. R., P. J. Phillips, D. S. Bolme, B. A. Draper, G. H. Givens, Y. M. Lui, M. N. Teli, H. Zhang, W. T. Scruggs, K. W. Bowyer, et al.**, The challenge of face recognition from digital point-and-shoot cameras. *In IEEE BTAS*. 2013.
19. **Bianchi, T.** and **A. Piva** (2012). Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, **7**(3), 1003–1017.
20. **Boyle, M., C. Edwards,** and **S. Greenberg**, The effects of filtered video on awareness and privacy. *In CSCW*. ACM, 2000.
21. **BuzzFeedVideo** (2018). You Won't Believe What Obama Says In This Video! <https://www.youtube.com/watch?v=cQ54GDm1eL0>.
22. **Cai, Z., Z. Gao, B. Planche, M. Zheng, T. Chen, M. S. Asif,** and **Z. Wu**, Disguise without disruption: Utility-preserving face de-identification. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38. 2024.
23. **Cao, J., C. Ma, T. Yao, S. Chen, S. Ding,** and **X. Yang**, End-to-end reconstruction-classification learning for face forgery detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022.
24. **Cao, Q., L. Shen, W. Xie, O. M. Parkhi,** and **A. Zisserman**, Vggface2: A dataset for recognising faces across pose and age. *In IEEE International Conference on Automatic Face & Gesture Recognition*. 2018.
25. **Carlini, N.** and **D. Wagner** (2016). Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*.
26. **Carlini, N.** and **D. Wagner**, Towards evaluating the robustness of neural networks. *In IEEE SP*. 2017.
27. **Carter, B., S. Jain, J. W. Mueller,** and **D. Gifford** (2021). Overinterpretation reveals image classification model pathologies. *Advances in Neural Information Processing Systems*, **34**.
28. **Chan, A., Y. Tay, Y. S. Ong,** and **J. Fu**, Jacobian adversarially regularized networks for robustness. *In International Conference on Learning Representations*. 2020. URL <https://openreview.net/forum?id=Hke0V1rKPS>.
29. **Chefer, H., I. Schwartz,** and **L. Wolf** (2022). Optimizing relevance maps of vision transformers improves robustness. *Advances in Neural Information Processing Systems*.

30. **Chen, J.-W., L.-J. Chen, C.-M. Yu, and C.-S. Lu**, Perceptual indistinguishability-net (pi-net): Facial image obfuscation with manipulable semantics. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2021a.
31. **Chen, Z., L. Xie, S. Pang, Y. He, and B. Zhang**, Magdr: Mask-guided detection and reconstruction for defending deepfakes. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021b.
32. **Chen, Z., T. Zhu, P. Xiong, C. Wang, and W. Ren** (2021c). Privacy preservation for image data: A gan-based method. *IJIS*, **36**(4), 1668–1685.
33. **Chhabra, S., P. Majumdar, M. Vatsa, and R. Singh**, Data fine-tuning. *In AAAI Conference on Artificial Intelligence*. 2019.
34. **Chhabra, S., R. Singh, M. Vatsa, and G. Gupta**, Anonymizing k-facial attributes via adversarial perturbations. *In International Joint Conference on Artificial Intelligence*. 2018.
35. **Choi, Y., M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo**, Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
36. **Chollet, F.**, Xception: Deep learning with depthwise separable convolutions. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
37. **Christlein, V., C. Riess, J. Jordan, C. Riess, and E. Angelopoulou** (2012). An evaluation of popular copy-move forgery detection approaches. *IEEE Transactions on Information Forensics and Security*, **7**(6), 1841–1854.
38. **Commision, N. P.** (2012). REPUBLIC ACT 10173 DATA PRIVACY ACT OF 2012. <https://privacy.gov.ph/data-privacy-act//>.
39. **Cootes, T. F., G. J. Edwards, and C. J. Taylor** (2001). Active appearance models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (6), 681–685.
40. **Dale, K., K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister**, Video face replacement. *In SIGGRAPH Asia Conference*. 2011.
41. **DeepFakes** (2017). Deepfakes FaceSwap. <https://tinyurl.com/y7gnkurs>.
42. **Dehghani, M., J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer, A. P. Steiner, M. Caron, R. Geirhos, I. Alabdulmohsin, et al.**, Scaling vision transformers to 22 billion parameters. *In International Conference on Machine Learning*. PMLR, 2023.
43. **Demontis, A., M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli**, Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks. *In 28th USENIX Conference on Security Symposium*. 2019.
44. **Deng, J., J. Guo, N. Xue, and S. Zafeiriou**, Arcface: Additive angular margin loss for deep face recognition. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

45. **Deng, Y., Y. Yang, B. Mirzasoleiman, and Q. Gu** (2024). Robust learning with progressive data expansion against spurious correlation. *Advances in neural information processing systems*, **36**.
46. **DeVries, T.** (2017). Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.
47. **Dias, Z., S. Goldenstein, and A. Rocha** (2013). Large-scale image phylogeny: Tracing image ancestral relationships. *IEEE Multimedia*, **20**(3), 58–70.
48. **Dias, Z., A. Rocha, and S. Goldenstein** (2011). Image phylogeny by minimal spanning trees. *IEEE Transactions on Information Forensics and Security*, **7**(2), 774–788.
49. **Ding, C. and D. Tao** (2018). Trunk-branch ensemble convolutional neural networks for video-based face recognition. *IEEE transactions on Pattern Analysis and Machine Intelligence*, **40**(4), 1002–1014.
50. **Ding, Z. and Y. Fu** (2017). Robust transfer metric learning for image classification. *IEEE Transactions on Image Processing*, **26**(2), 660–670.
51. **Dogra, V., S. Verma, M. Woźniak, J. Shafi, M. F. Ijaz, et al.** (2024). Shortcut learning explanations for deep natural language processing: A survey on dataset biases. *IEEE Access*.
52. **Dong, S., J. Wang, R. Ji, J. Liang, H. Fan, and Z. Ge**, Implicit identity leakage: The stumbling block to improving deepfake detection generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
53. **Du, M., N. Liu, F. Yang, and X. Hu** (2020). Learning credible dnns via incorporating prior knowledge and model local explanation. *Knowledge and Information Systems*, 1–28.
54. **Ehrlich, M., T. J. Shields, T. Almaev, and M. R. Amer**, Facial attributes classification using multi-task representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016.
55. **Em, Y., F. Gag, Y. Lou, S. Wang, T. Huang, and L.-Y. Duan**, Incorporating intra-class variance to fine-grained visual recognition. In *IEEE International Conference on Multimedia and Expo*. 2017.
56. **Erichson, B., S. H. Lim, W. Xu, F. Utrera, Z. Cao, and M. Mahoney**, Noisymix: Boosting model robustness to common corruptions. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024.
57. **Fan, Y., X. Lu, D. Li, and Y. Liu**, Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In *18th ACM International Conference on Multimodal Interaction*. 2016.
58. **Fan, Z., S. Hu, J. Yao, G. Niu, Y. Zhang, M. Sugiyama, and Y. Wang** (2024). Locally estimated global perturbations are better than local perturbations for federated sharpness-aware minimization. *arXiv preprint arXiv:2405.18890*.

59. **Fang, Y., W. Wang, B. Xie, Q. Sun, L. Wu, X. Wang, T. Huang, X. Wang, and Y. Cao**, Eva: Exploring the limits of masked visual representation learning at scale. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
60. **Fawzi, A., S.-M. Moosavi-Dezfooli, P. Frossard, and S. Soatto**, Empirical study of the topology and geometry of deep networks. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
61. **Fernando, T., C. Fookes, S. Denman, and S. Sridharan** (2019). Exploiting human social cognition for the detection of fake and fraudulent faces via memory networks. *arXiv preprint arXiv:1911.07844*.
62. **Gafni, O., L. Wolf, and Y. Taigman**, Live face de-identification in video. *In IEEE International Conference on Computer Vision*. 2019.
63. **Gao, C., H. Zhou, J. Yu, Y. Ye, J. Cai, J. Wang, and W. Yang**, Attacking transformers with feature diversity adversarial perturbation. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38. 2024.
64. **Gao, J., J. Zhang, X. Liu, T. Darrell, E. Shelhamer, and D. Wang**, Back to the source: Diffusion-driven adaptation to test-time corruption. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
65. **Geirhos, R., J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann** (2020). Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11), 665–673.
66. **Gnanasekar, S. T. and S. Yanushkevich**, Face attributes and detection of drug addicts. *In IEEE EST*. 2019.
67. **Goel, A., A. Singh, A. Agarwal, M. Vatsa, and R. Singh** (2018). Smartbox: Benchmarking adversarial detection and mitigation algorithms for face recognition. *IEEE International Conference on Biometrics: Theory, Applications, and Systems*.
68. **Goodfellow, I. J., J. Shlens, and C. Szegedy** (2014). Explaining and harnessing adversarial examples. *International Conference on Learning Representation*.
69. **Goswami, G., N. Ratha, A. Agarwal, R. Singh, and M. Vatsa**, Unravelling robustness of deep learning based face recognition against adversarial attacks. *In Association for the Advancement of Artificial Intelligence*. 2018.
70. **Goswami, G., M. Vatsa, and R. Singh** (2017). Face verification via learned representation on feature-rich video frames. *IEEE TIFS*, 12(7), 1686–1698.
71. **Gross, R., E. Airoidi, B. Malin, and L. Sweeney**, Integrating utility into face de-identification. *In International Workshop on Privacy Enhancing Technologies*. Springer, 2005.
72. **Gross, R., L. Sweeney, F. De la Torre, and S. Baker**, Model-based face de-identification. *In IEEE CVPRW*. 2006.
73. **Gu, X., W. Luo, M. S. Ryoo, and Y. J. Lee**, Password-conditioned anonymization and deanonymization with face identity transformers. *In European Conference on Computer Vision*. Springer, 2020.

74. **Guo, Y., D. Stutz, and B. Schiele**, Improving robustness of vision transformers by reducing sensitivity to patch corruptions. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
75. **Han, H., A. K. Jain, F. Wang, S. Shan, and X. Chen** (2017a). Heterogeneous face attribute estimation: A deep multi-task learning approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **40**(11), 2597–2609.
76. **Han, X., V. Morariu, P. I. Larry Davis, et al.**, Two-stream neural networks for tampered face detection. *In IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017b.
77. **Hand, E. M., C. D. Castillo, and R. Chellappa**, Doing the best we can with what we have: Multi-label balancing with selective learning for attribute prediction. *In Association for the Advancement of Artificial Intelligence*. 2018.
78. **Hand, E. M. and R. Chellappa**, Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification. *In AAAI Conference on Artificial Intelligence*. 2017.
79. **He, K., Y. Fu, W. Zhang, C. Wang, Y.-G. Jiang, F. Huang, and X. Xue**, Harnessing synthesized abstraction images to improve facial attribute recognition. *In IJCAI*. 2018.
80. **He, K., X. Zhang, S. Ren, and J. Sun**, Deep residual learning for image recognition. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
81. **He, X., M. Zhu, D. Chen, N. Wang, and X. Gao** (2024). Diff-privacy: Diffusion-based face privacy protection. *IEEE Transactions on Circuits and Systems for Video Technology*.
82. **Hendrycks, D., S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, et al.**, The many faces of robustness: A critical analysis of out-of-distribution generalization. *In Proceedings of the IEEE International Conference on Computer Vision*. 2021.
83. **Hendrycks, D. and T. Dietterich** (2019a). Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*.
84. **Hendrycks, D. and T. Dietterich**, Benchmarking neural network robustness to common corruptions and perturbations. *In International Conference on Learning Representations*. 2019b. URL <https://openreview.net/forum?id=HJz6tiCqYm>.
85. **Hendrycks, D. and T. G. Dietterich** (2018). Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*.
86. **Hendrycks, D., N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan** (2019). Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*.
87. **Hendrycks, D., A. Zou, M. Mazeika, L. Tang, B. Li, D. Song, and J. Steinhardt**, Pixmix: Dreamlike pictures comprehensively improve safety measures. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

88. **Hill, K.** (2020). Wrongfully Accused by an Algorithm. <https://www.nytimes.com/2020/06/24/technology/facial-recognition-arrest.html>.
89. **Hoffman, J., D. A. Roberts, and S. Yaida** (2019). Robust learning with jacobian regularization. *arXiv preprint arXiv:1908.02729*.
90. **Huang, G. B., M. Mattar, T. Berg, and E. Learned-Miller**, Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *In Workshop on Faces in 'Real-Life' Images: Detection, Alignment, and Recognition*. 2008.
91. **Huang, G. B., M. Ramesh, T. Berg, and E. Learned-Miller** (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.
92. **Huang, Z., X. Bao, N. Zhang, Q. Zhang, X. Tu, B. Wu, and X. Yang** (2023). Ipmix: Label-preserving data augmentation method for training robust classifiers. *Advances in Neural Information Processing Systems*, **36**, 63660–63673.
93. **Hukkelås, H., R. Mester, and F. Lindseth**, Deepprivacy: A generative adversarial network for face anonymization. *In International symposium on visual computing*. Springer, 2019.
94. **Ilyas, A., S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry**, Adversarial examples are not bugs, they are features. *In Advances in Neural Information Processing Systems*. 2019.
95. **Intelligence, A.** (2024). Medical ai models rely on shortcuts that could lead to misdiagnosis of covid-19. <https://www.itnonline.com/content/medical-ai-models-rely-shortcuts-could-lead-misdiagnosis-covid-19>.
96. **Islam, K., M. Z. Zaheer, A. Mahmood, and K. Nandakumar**, Diffusemix: Label-preserving data augmentation with diffusion models. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024a.
97. **Islam, K., M. Z. Zaheer, A. Mahmood, K. Nandakumar, and N. Akhtar** (2024b). Genmix: Effective data augmentation with generative diffusion model image editing. *arXiv preprint arXiv:2412.02366*.
98. **Jain, Y. K. and S. K. Bhandare** (2011). Min max normalization based data perturbation method for privacy protection. *International Journal of Computer & Communication Technology*, **2**(8), 45–50.
99. **Jiang, B., B. Bai, H. Lin, Y. Wang, Y. Guo, and L. Fang**, Dartblur: Privacy preservation with detection artifact suppression. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
100. **Jin, X., H. Zhu, S. Li, Z. Wang, Z. Liu, C. Yu, H. Qin, and S. Z. Li** (2024). A survey on mixup augmentations and beyond. *arXiv preprint arXiv:2409.05202*.
101. **Jourabloo, A., X. Yin, and X. Liu**, Attribute preserved face de-identification. *In IEEE ICB*. 2015.
102. **Ju, Y., S. Hu, S. Jia, G. H. Chen, and S. Lyu**, Improving fairness in deepfake detection. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024.

103. **Jung, H., S. Lee, J. Yim, S. Park, and J. Kim**, Joint fine-tuning in deep neural networks for facial expression recognition. *In IEEE International Conference on Computer Vision*. 2015.
104. **Kalayeh, M. M., B. Gong, and M. Shah** (2017). Improving facial attribute prediction using semantic segmentation. *arXiv preprint arXiv:1704.08740*.
105. **Kapoor, S.** (2024). How has social media contributed to the spread of cybercrimes : an analysis. <https://blog.ipleaders.in/how-has-social-media-contributed-to-the-spread-of-cybercrimes-an-analysis/>.
106. **Karras, T., T. Aila, S. Laine, and J. Lehtinen**, Progressive growing of GANs for improved quality, stability, and variation. *In International Conference on Learning Representations*. 2018. URL <https://openreview.net/forum?id=Hk99zCeAb>.
107. **Kastrenakes, J.** (2019). Controversial deepfake app DeepNude shuts down hours after being exposed. <https://tinyurl.com/yxe2th8j>.
108. **Kazemi, H., S. Soleymani, A. Dabouei, M. Iranmanesh, and N. M. Nasrabadi**, Attribute-centered loss for soft-biometrics guided face sketch-photo recognition. *In IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018.
109. **Kim, H., P. Garrido, A. Tewari, W. Xu, J. Thies, M. Nießner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt** (2018). Deep video portraits. *ACM Transactions on Graphics*, **37**(4), 1–14.
110. **Korshunov, P. and T. Ebrahimi**, Using face morphing to protect privacy. *In IEEE International Conference on Advanced Video and Signal Based Surveillance*. 2013.
111. **Korshunov, P. and T. Ebrahimi**, Towards optimal distortion-based visual privacy filters. *In IEEE International Conference on Image Processing*. 2014.
112. **Krizhevsky, A., G. Hinton, et al.** (2009). Learning multiple layers of features from tiny images.
113. **Krizhevsky, A., I. Sutskever, and G. E. Hinton**, Imagenet classification with deep convolutional neural networks. *In Advances in Neural Information Processing Systems*. 2012.
114. **Krueger, D., E. Caballero, J.-H. Jacobsen, A. Zhang, J. Binas, D. Zhang, R. Le Priol, and A. Courville**, Out-of-distribution generalization via risk extrapolation (rex). *In International Conference on Machine Learning*. PMLR, 2021.
115. **Kuang, Z., J. Lu, C. Hong, H. Huang, S. Zhu, X. Zhao, J. Yu, and J. Fan**, Latent representation reorganization for face privacy protection. *In ACM Multimedia*. 2024a.
116. **Kuang, Z., X. Yang, Y. Shen, C. Hu, and J. Yu**, Facial identity anonymization via intrinsic and extrinsic attention distraction. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024b.
117. **Kumar, N., A. C. Berg, P. N. Belhumeur, and S. K. Nayar**, Attribute and simile classifiers for face verification. *In IEEE International Conference on Computer Vision*. 2009.

118. **Kumar, P., M. Vatsa, and R. Singh**, Detecting face2face facial reenactment in videos. *In IEEE Winter Conference on Applications of Computer Vision*. 2020.
119. **Kundu, A., V. Vineet, and V. Koltun**, Feature space optimization for semantic video segmentation. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
120. **Kurakin, A., I. Goodfellow, and S. Bengio** (2016). Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.
121. **Kwon, J., E. Lee, Y. Cho, and Y. Kim**, Learning to detour: Shortcut mitigating augmentation for weakly supervised semantic segmentation. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024.
122. **Laishram, L., J. T. Lee, and S. K. Jung** (2024). Face de-identification using face caricature. *IEEE Access*.
123. **Lapuschkin, S., S. Wäldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Müller** (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, **10**(1), 1–8.
124. **Last, M., T. Tassa, A. Zhmudyak, and E. Shmueli** (2014). Improving accuracy of classification models induced from anonymized datasets. *Information Sciences*, **256**, 138–161.
125. **Le, B. M. and S. S. Woo**, Quality-agnostic deepfake detection with intra-model collaborative learning. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.
126. **Le, M.-H., M. S. N. Khan, G. Tsaloli, N. Carlsson, and N. Buchegger**, Anonfaces: Anonymizing faces adjusted to constraints on efficacy and security. *In ACM Workshop on Privacy in the Electronic Society*. 2020.
127. **Le, Y. and X. Yang** (2015). Tiny imagenet visual recognition challenge. *CS 231N*, **7**(7), 3.
128. **LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner** (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86**(11), 2278–2324.
129. **Li, C., C. Xu, and Z. Feng** (2016). Analysis of physiological for emotion recognition with the irs model. *Neurocomputing*, **178**, 103–111.
130. **Li, D., W. Wang, K. Zhao, J. Dong, and T. Tan**, Riddle: Reversible and diversified de-identification with latent encryptor. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023a.
131. **Li, J., L. Han, R. Chen, H. Zhang, B. Han, L. Wang, and X. Cao**, Identity-preserving face anonymization via adaptively facial attributes obfuscation. *In Proceedings of the 29th ACM international conference on multimedia*. 2021.
132. **Li, J., X. Li, B. Yang, and X. Sun** (2014). Segmentation-based image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security*, **10**(3), 507–518.

133. **Li, J., H. Zhang, S. Liang, P. Dai, and X. Cao** (2023b). Privacy-enhancing face obfuscation guided by semantic-aware attribution maps. *IEEE Transactions on Information Forensics and Security*, **18**, 3632–3646.
134. **Li, T. and L. Lin**, Anonymousnet: Natural face de-identification with measurable privacy. *In IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2019.
135. **Li, X. and Z. Zhou**, Secure support vector machines with data perturbation. *In Chinese Control And Decision Conference*. 2018.
136. **Li, Y., M.-C. Chang, and S. Lyu**, In ictu oculi: Exposing ai created fake videos by detecting eye blinking. *In IEEE International Workshop on Information Forensics and Security*. 2018.
137. **Li, Y. and S. Lyu** (2018). Exposing deepfake videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656*.
138. **Li, Y., X. Yang, P. Sun, H. Qi, and S. Lyu** (2019). Celeb-df: A new dataset for deepfake forensics. *arXiv preprint arXiv:1909.12962*.
139. **Li, Y., X. Yang, P. Sun, H. Qi, and S. Lyu**, Celeb-df: A large-scale challenging dataset for deepfake forensics. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2020.
140. **Li, Z., I. Evtimov, A. Gordo, C. Hazirbas, T. Hassner, C. C. Ferrer, C. Xu, and M. Ibrahim**, A whac-a-mole dilemma: Shortcuts come in multiples where mitigating one amplifies others. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023c.
141. **Lin, J., Y. Li, and G. Yang** (2021). Fpgan: Face de-identification method with generative adversarial networks for social robots. *Elsevier NN*, **133**, 132–147.
142. **Ling, T., S. Siping, H. Wang, C. Hu, and D. Wang**, Federated morozov regularization for shortcut learning in privacy preserving learning with watermarked image data. *In ACM Multimedia 2024*. 2024.
143. **Liu, H., X. Li, W. Zhou, Y. Chen, Y. He, H. Xue, W. Zhang, and N. Yu**, Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021.
144. **Liu, M., Y. Ding, M. Xia, X. Liu, E. Ding, W. Zuo, and S. Wen**, Stgan: A unified selective transfer network for arbitrary image attribute editing. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
145. **Liu, X., Y. Zhong, W. Deng, H. Shi, X. Cui, Y. Yin, and D. Wen**, Enhancing generalization of invisible facial privacy cloak via gradient accumulation. *In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.
146. **Liu, Z., P. Luo, X. Wang, and X. Tang**, Deep learning face attributes in the wild. *In ICCV*. 2015a.

147. **Liu, Z., P. Luo, X. Wang, and X. Tang**, Deep learning face attributes in the wild. *In IEEE International Conference on Computer Vision*. 2015b.
148. **Lopez, J., C. Hinojosa, H. Arguello, and B. Ghanem**, Privacy-preserving optics for enhancing protection in face de-identification. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.
149. **Machireddy, A., R. Krishnan, N. Ahuja, and O. Tickoo**, Continual active adaptation to evolving distributional shifts. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2022.
150. **Mao, L., Y. Yan, J.-H. Xue, and H. Wang** (2020). Deep multi-task multi-label cnn for effective facial attribute classification. *IEEE Transactions on Affective Computing*.
151. **MarekKowalski** (2016). FaceSwap. <https://github.com/MarekKowalski/FaceSwap/>.
152. **Maximov, M., I. Elezi, and L. Leal Taixe**, Ciagan: Conditional identity anonymization generative adversarial networks. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2020.
153. **Meden, B., R. C. Malli, S. Fabijan, H. K. Ekenel, V. Štruc, and P. Peer** (2017). Face deidentification with generative deep neural networks. *IET Signal Processing*, **11**(9), 1046–1054.
154. **Mehra, A., A. Agarwal, M. Vatsa, and R. Singh** (2023). Motion magnified 3d residual-in-dense network for deepfake detection. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, **5**(1), 39–52.
155. **Milborrow, S., J. Morkel, and F. Nicolls** (2010a). The MUCT Landmarked Face Database. *PRASA*. <http://www.milbo.org/muct>.
156. **Milborrow, S., J. Morkel, and F. Nicolls** (2010b). The MUCT landmarked face database. *Pattern Recognition Association of South Africa*, **201**(0).
157. **Mirjalili, V., S. Raschka, A. Namboodiri, and A. Ross** (2017). Semi-adversarial networks: Convolutional autoencoders for imparting privacy to face images. *arXiv preprint arXiv:1712.00321*.
158. **Mirjalili, V., S. Raschka, and A. Ross**, Gender privacy: An ensemble of semi adversarial networks for confounding arbitrary gender classifiers. *In IEEE International Conference on Biometrics Theory, Applications and Systems*. 2018.
159. **Mirjalili, V., S. Raschka, and A. Ross** (2020). Privacynet: Semi-adversarial networks for multi-attribute face privacy. *IEEE Transactions on Image Processing*, **29**, 9400–9412.
160. **Mirjalili, V. and A. Ross** (2017). Soft biometric privacy: Retaining biometric utility of face images while perturbing gender. *IEEE IJCB*.
161. **Mittal, A., R. Soundararajan, and A. C. Bovik** (2012). Making a “completely blind” image quality analyzer. *IEEE Signal processing letters*, **20**(3), 209–212.
162. **Mittal, T., U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha**, Emotions don’t lie: An audio-visual deepfake detection method using affective cues. *In Proceedings of ACM International Conference on Multimedia*. 2020.

163. **Moosavi-Dezfooli, S.-M., A. Fawzi, O. Fawzi, and P. Frossard**, Universal adversarial perturbations. *In Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.
164. **Moosavi-Dezfooli, S.-M., A. Fawzi, and P. Frossard**, Deepfool: a simple and accurate method to fool deep neural networks. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
165. **Newton, E. M., L. Sweeney, and B. Malin** (2005). Preserving privacy by de-identifying face images. *IEEE TKDE*, **17**(2), 232–243.
166. **Ng, T.-T. and S.-F. Chang**, A model for image splicing. *In International Conference on Image Processing*, volume 2. 2004.
167. **Nguyen, H. H., F. Fang, J. Yamagishi, and I. Echizen** (2019a). Multi-task learning for detecting and segmenting manipulated facial images and videos. *arXiv preprint arXiv:1906.06876*.
168. **Nguyen, H. H., J. Yamagishi, and I. Echizen**, Capsule-forensics: Using capsule networks to detect forged images and videos. *In IEEE International Conference on Acoustics, Speech and Signal Processing*. 2019b.
169. **Oakden-Rayner, L., J. Dunnmon, G. Carneiro, and C. Ré**, Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. *In Proceedings of the ACM Conference on Health, Inference, and Learning*. 2020.
170. **Othman, A. A. and A. Ross**, Privacy of facial soft biometrics: Suppressing gender but retaining identity. *In ECCV Workshops (2)*. 2014.
171. **Pan, S. J., Q. Yang, et al.** (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, **22**(10), 1345–1359.
172. **Pan, Y., J. Chen, and J. Wu**, A multi-factor combinations enhanced reversible privacy protection system for facial images. *In ICME*. 2021.
173. **Pandey, M.** (2022). Epic AI Fails — A List of Failed Machine Learning Projects. <https://analyticsindiamag.com/epic-ai-fails-a-list-of-failed-machine-learning-projects/>.
174. **Papernot, N., P. McDaniel, and I. Goodfellow** (2016a). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*.
175. **Papernot, N., P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami**, The limitations of deep learning in adversarial settings. *In IEEE European Symposium on Security and Privacy*. 2016b.
176. **Parkhi, O. M., A. Vedaldi, A. Zisserman, et al.**, Deep face recognition. *In BMVC*, volume 1. 2015.
177. **Pellicer, A. L., Y. Li, and P. Angelov**, Pudd: Towards robust multi-modal prototype-based deepfake detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.

178. **Pettit, H.** (2017). Do you look like a politician, porn star or golfer? algorithm reveals the 'average face' of people in different professions. <https://www.dailymail.co.uk/sciencetech/article-4951430/Algorithm-reveals-average-face-different-professions.html>.
179. **Pezeshki, M., O. Kaba, Y. Bengio, A. C. Courville, D. Precup, and G. Lajoie** (2021). Gradient starvation: A learning proclivity in neural networks. *Advances in Neural Information Processing Systems*, **34**.
180. **Poole, B., S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli**, Exponential expressivity in deep neural networks through transient chaos. *In Advances in Neural Information Processing Systems*. 2016.
181. **Qi, C. and F. Su**, Contrastive-center loss for deep neural networks. *In IEEE International Conference on Image Processing*. 2017.
182. **Qian, Y., G. Yin, L. Sheng, Z. Chen, and J. Shao**, Thinking in frequency: Face forgery detection by mining frequency-aware clues. *In European Conference on Computer Vision*. Springer, 2020.
183. **Rahmouni, N., V. Nozick, J. Yamagishi, and I. Echizen**, Distinguishing computer graphics from natural images using convolution neural networks. *In IEEE Workshop on Information Forensics and Security*. 2017.
184. **Rana, K., A. Pandey, P. Goyal, G. Singh, and P. Goyal** (2023). A novel privacy protection approach with better human imperceptibility: K. rana et al. *Applied Intelligence*, **53**(19), 21788–21798.
185. **Rana, M. S., M. N. Nobi, B. Murali, and A. H. Sung** (2022). Deepfake detection: a systematic literature review. *IEEE Access*.
186. **Rose, J. and T. Bourlai**, Deep learning based estimation of facial attributes on challenging mobile phone face datasets. *In ACM International Conference on Advances in Social Networks Analysis and Mining*. 2019.
187. **Ross, A. and F. Doshi-Velez**, Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. 2018.
188. **Rössler, A., D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner** (2018). Faceforensics: A large-scale video dataset for forgery detection in human faces. *arXiv preprint arXiv:1803.09179*.
189. **Rössler, A., D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner**, Faceforensics++: Learning to detect manipulated facial images. *In IEEE International Conference on Computer Vision*. 2019.
190. **Rot, P., J. Križaj, P. Peer, and V. Štruc**, Enhancing gender privacy with photo-realistic fusion of disentangled spatial segments. *In ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.
191. **Rozsa, A., M. Günther, E. M. Rudd, and T. E. Boulton**, Are facial attributes adversarially robust? *In IEEE ICPR*. 2016.

192. **Rozsa, A., M. Günther, E. M. Rudd, and T. E. Boulton** (2017). Facial attributes: Accuracy and adversarial robustness. *PRL* <https://doi.org/10.1016/j.patrec.2017.10.024>.
193. **Rozsa, A., M. Günther, E. M. Rudd, and T. E. Boulton** (2019). Facial attributes: Accuracy and adversarial robustness. *Pattern Recognition Letters*, **124**, 100–108.
194. **Rusak, E., L. Schott, R. S. Zimmermann, J. Bitterwolf, O. Bringmann, M. Bethge, and W. Brendel**, A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision*. Springer, 2020.
195. **Sabir, E., J. Cheng, A. Jaiswal, W. AbdAlmageed, I. Masi, and P. Natarajan** (2019). Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces*, **3**, 1.
196. **Salamon, J. and J. P. Bello** (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, **24**(3), 279–283.
197. **Samarati, P. and L. Sweeney** (1998). Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression.
198. **Saxen, F., P. Werner, S. Handrich, E. Othman, L. Dinges, and A. Al-Hamadi**, Face attribute detection with mobilenetv2 and nasnet-mobile. In *IEEE International Symposium on Image and Signal Processing and Analysis*. 2019.
199. **Sayles, A., A. Hooda, M. Gupta, R. Chatterjee, and E. Fernandes**, Invisible perturbations: Physical adversarial examples exploiting the rolling shutter effect. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2021.
200. **Schneider, S., E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge** (2020). Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, **33**.
201. **Schroff, F., D. Kalenichenko, and J. Philbin**, Facenet: A unified embedding for face recognition and clustering. In *IEEE CVPR*. 2015.
202. **Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra**, Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*. 2017.
203. **Sethi, A., M. Singh, R. Singh, and M. Vatsa** (2018). Residual codean autoencoder for facial attribute analysis. *PRL*.
204. **Sethi, A., M. Singh, R. Singh, and M. Vatsa** (2019). Residual codean autoencoder for facial attribute analysis. *Pattern Recognition Letters*, **119**, 157–165.
205. **Shamshad, F., M. Naseer, and K. Nandakumar**, Clip2protect: Protecting facial privacy using text-guided makeup via adversarial latent search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
206. **Shao, R., T. Wu, L. Nie, and Z. Liu** (2025). Deepfake-adapter: Dual-level adapter for deepfake detection. *International Journal of Computer Vision*, 1–16.
207. **Shen, W. and R. Liu**, Learning residual images for face attribute manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

208. **Shen, Z., S. Fan, Y. Wong, T.-T. Ng, and M. Kankanhalli**, Human-imperceptible privacy protection against machines. *In Proceedings of the 27th ACM international conference on multimedia*. 2019.
209. **Shi, H., X. Zhu, Z. Lei, S. Liao, and S. Z. Li**, Learning discriminative features with class encoder. *In IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2016.
210. **Sim, T. and L. Zhang**, Controllable face privacy. *In IEEE AFGR*, volume 4. 2015.
211. **Singh, A., E. Garza, A. Chopra, P. Vepakomma, V. Sharma, and R. Raskar** (2022). Decouple-and-sample: Protecting sensitive information in task agnostic data release. *arXiv preprint arXiv:2203.13204*.
212. **Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov** (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, **15**(1), 1929–1958.
213. **Strisciuglio, N. and G. Azzopardi**, Visual response inhibition for increased robustness of convolutional networks to distribution shifts. *In NeurIPS Workshop on Distribution Shifts: Connecting Methods and Applications*. 2022.
214. **Su, J., D. V. Vargas, and S. Kouichi** (2017). One pixel attack for fooling deep neural networks. *arXiv preprint arXiv:1710.08864*.
215. **Su, W., X. Zhu, C. Tao, L. Lu, B. Li, G. Huang, Y. Qiao, X. Wang, J. Zhou, and J. Dai**, Towards all-in-one pre-training via maximizing multi-modal mutual information. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023.
216. **Sun, Z., Y. Han, Z. Hua, N. Ruan, and W. Jia**, Improving the efficiency and robustness of deepfakes detection through precise geometric features. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
217. **Suo, J., L. Lin, S. Shan, X. Chen, and W. Gao** (2011). High-resolution face fusion for gender conversion. *IEEE TSMC*, **41**(2), 226–237.
218. **Sweeney, L.** (2002). k-anonymity: A model for protecting privacy. *IJUFKS*, **10**(05), 557–570.
219. **Szegedy, C., W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus** (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.
220. **Taesiri, M. R., G. Nguyen, S. Habchi, C.-P. Bezemer, and A. Nguyen** (2024). Imagenet-hard: The hardest images remaining from a study of the power of zoom and spatial biases in image classification. *Advances in Neural Information Processing Systems*, **36**.
221. **Tan, C., Y. Zhao, S. Wei, G. Gu, P. Liu, and Y. Wei**, Rethinking the up-sampling operations in cnn-based generative network for generalizable deepfake detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.

222. **Tan, M.** and **Q. Le**, Efficientnet: Rethinking model scaling for convolutional neural networks. *In International conference on machine learning*. PMLR, 2019.
223. **Tan, M., R. Pang,** and **Q. V. Le**, Efficientdet: Scalable and efficient object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020.
224. **Taori, R., A. Dave, V. Shankar, N. Carlini, B. Recht,** and **L. Schmidt** (2020). Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, **33**.
225. **Korshunov, P.** and **T. Ebrahimi**, Using warping for privacy protection in video surveillance. *In International Conference on Digital Signal Processing*. 2013.
226. **Thakral, K., S. Prasad, S. Aswani, M. Vatsa,** and **R. Singh**, Toonergan: Reinforcing gans for obfuscating automated facial indexing. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2024.
227. **Thies, J., M. Zollhöfer,** and **M. Nießner** (2019). Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics*, **38**(4), 1–12.
228. **Thies, J., M. Zollhofer, M. Stamminger, C. Theobalt,** and **M. Nießner**, Face2face: Real-time face capture and reenactment of rgb videos. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2016.
229. **Thom, N.** and **E. M. Hand** (2020). Facial attribute recognition: A survey. *Computer Vision: A Reference Guide*, 1–13.
230. **Timpl, L., R. Entezari, H. Sedghi, B. Neyshabur,** and **O. Saukh** (2022). Understanding the effect of sparsity on neural networks robustness. *arXiv preprint arXiv:2206.10915*.
231. **Um, T. T., F. M. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek,** and **D. Kulić**, Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks. *In ACM International Conference on Multimodal Interaction*. 2017.
232. **University, Y.** (2022). Ai use potentially dangerous “shortcuts” to solve complex recognition tasks. <https://tinyurl.com/4873amzm>.
233. **Varga, D., A. Csiszárík,** and **Z. Zombori** (2017). Gradient regularization improves accuracy of discriminative models. *arXiv preprint arXiv:1712.09936*.
234. **Verdoliva, L.** (2020). Media forensics and deepfakes: an overview. *IEEE Journal of Selected Topics in Signal Processing*, **14**(5), 910–932.
235. **Viola, P.** and **M. Jones**, Rapid object detection using a boosted cascade of simple features. *In IEEE Conference on Computer Vision and Pattern Recognition*, volume 1. 2001.
236. **Viola, P., M. J. Jones,** and **D. Snow** (2005). Detecting pedestrians using patterns of motion and appearance. *IJCV*, **63**(2), 153–161.

237. **Wang, C., G. Zhang, and R. Grosse**, Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*. 2020. URL <https://openreview.net/forum?id=SkgsACVKPH>.
238. **Wang, H., Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu**, Cosface: Large margin cosine loss for deep face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
239. **Wang, S., C. Brune, R. Veldhuis, and N. Strisciuglio (2023a)**. Dfm-x: Augmentation by leveraging prior knowledge of shortcut learning. *arXiv preprint arXiv:2308.06622*.
240. **Wang, T., X. Liao, K. P. Chow, X. Lin, and Y. Wang (2024)**. Deepfake detection: A comprehensive survey from the reliability perspective. *ACM Computing Surveys*, **57**(3), 1–35.
241. **Wang, W., J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, et al.**, Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023b.
242. **Wang, Y. and A. Dantcheva**, A video is worth more than 1000 lies. comparing 3dcnn approaches for detecting deepfakes. In *2020 15Th IEEE international conference on automatic face and gesture recognition (FG 2020)*. IEEE, 2020.
243. **Wang, Y. and M. Kosinski (2017)**. Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. *PsyArXiv preprint arXiv:10.17605/OSF.IO/HV28A*.
244. **Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli (2004)**. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, **13**(4), 600–612.
245. **Wei, X., Y. Huang, Y. Sun, and J. Yu**, Unified adversarial patch for cross-modal attacks in the physical world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.
246. **Wen, Y., B. Liu, J. Cao, R. Xie, and L. Song**, Divide and conquer: a two-step method for high quality face de-identification with model explainability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.
247. **Wen, Y., B. Liu, J. Cao, R. Xie, L. Song, and Z. Li (2022)**. Identitymask: Deep motion flow guided reversible face video de-identification. *IEEE Transactions on Circuits and Systems for Video Technology*.
248. **Wen, Y., K. Zhang, Z. Li, and Y. Qiao**, A discriminative feature learning approach for deep face recognition. In *European Conference on Computer Vision*. Springer, 2016.
249. **Wilkin, J. (2018a)**. A Social Media Mapping Tool that correlates profiles via facial recognition. https://github.com/Greenwolf/social_mapper.
250. **Wilkin, J. (2018b)**. A social media mapping tool that correlates profiles via facial recognition. https://github.com/Greenwolf/social_mapper/.
251. **Wolf, L., T. Hassner, and I. Maoz**, Face recognition in unconstrained videos with matched background similarity. In *IEEE CVPR*. 2011.

252. **Wu, E., K. Wu, D. Cox, and W. Lotter** (2018a). Conditional infilling gans for data augmentation in mammogram classification. *arXiv preprint arXiv:1807.08093*.
253. **Wu, L., Z. Zhu, C. Tai, et al.** (2018b). Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*.
254. **Wu, X., R. He, Z. Sun, and T. Tan** (2018c). A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, **13**(11), 2884–2896.
255. **Wu, X., Z. Xie, Y. Gao, and Y. Xiao**, Sstnet: Detecting manipulated faces through spatial, steganalysis and temporal features. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2020.
256. **Wu, Y., F. Yang, and H. Ling** (2018d). Privacy-protective-gan for face de-identification. *arXiv preprint arXiv:1806.08906*.
257. **Xiao, D., J. Xia, M. Li, and M. Zhang** (2024). Manipulable, reversible and diversified de-identification via face identity disentanglement. *Multimedia Tools and Applications*, 1–18.
258. **Xiao, T., Y.-H. Tsai, K. Sohn, M. Chandraker, and M.-H. Yang**, Adversarial learning of privacy-preserving and task-oriented representations. In *AAAI Conference on Artificial Intelligence*, volume 34. 2020.
259. **Xie, C., J. Wang, Z. Zhang, Z. Ren, and A. Yuille** (2017). Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*.
260. **Xie, C., Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille**, Improving transferability of adversarial examples with input diversity. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
261. **Xie, Y., J. Zhang, H. Lu, C. Shen, and Y. Xia** (2020). Sesv: Accurate medical image segmentation by predicting and correcting errors. *IEEE Transactions on Medical Imaging*, **40**(1), 286–296.
262. **Yan, Z., Y. Zhang, Y. Fan, and B. Wu**, Ucf: Uncovering common features for generalizable deepfake detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.
263. **Yang, H., X. Xu, C. Xu, H. Zhang, J. Qin, Y. Wang, P.-A. Heng, and S. He** (2024a). G 2 face: High-fidelity reversible face anonymization via generative and geometric priors. *IEEE Transactions on Information Forensics and Security*.
264. **Yang, X., Y. Dong, T. Pang, J. Zhu, and H. Su** (2020). Towards privacy protection by generating adversarial identity masks. *arXiv preprint arXiv:2003.06814*.
265. **Yang, Y., E. Gan, G. K. Dziugaite, and B. Mirzasoleiman**, Identifying spurious biases early in training through the lens of simplicity bias. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024b.
266. **Yang, Z., J. Liang, Y. Xu, X.-Y. Zhang, and R. He** (2023). Masked relation learning for deepfake detection. *IEEE Transactions on Information Forensics and Security*, **18**, 1696–1708.

267. **Yang, Z., Y. Zhang, T. Wang, Z. Hua, Z. Xia, and J. Weng**, Once-for-all: Efficient visual face privacy protection via person-specific veils. *In ACM Multimedia*. 2024c.
268. **Yao, Z., A. Gholami, P. Xu, K. Keutzer, and M. W. Mahoney**, Trust region based adversarial attack on neural networks. *In IEEE Conference on Computer Vision and Pattern Recognition*. 2019.
269. **Yi, D., Z. Lei, S. Liao, and S. Z. Li** (2014). Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*.
270. **Yuan, L., W. Chen, X. Pu, Y. Zhang, H. Li, Y. Zhang, X. Gao, and T. Ebrahimi** (2024). Pro-face c: Privacy-preserving recognition of obfuscated face via feature compensation. *IEEE Transactions on Information Forensics and Security*.
271. **Yun, S., D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo**, Cutmix: Regularization strategy to train strong classifiers with localizable features. *In Proceedings of the IEEE/CVF international conference on computer vision*. 2019.
272. **Zhang, H., M. Cisse, Y. N. Dauphin, and D. Lopez-Paz**, mixup: Beyond empirical risk minimization. *In International Conference on Learning Representations*. 2018. URL <https://openreview.net/forum?id=r1Ddp1-Rb>.
273. **Zhang, J., H. Chao, A. Dhurandhar, P.-Y. Chen, A. Tajer, Y. Xu, and P. Yan**, When neural networks fail to generalize? a model sensitivity perspective. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37. 2023.
274. **Zhang, K., Z. Zhang, Z. Li, and Y. Qiao** (2016). Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, **23**(10), 1499–1503.
275. **Zhang, L., J. Yang, and D. Zhang** (2017). Domain class consistency based transfer learning for image classification across domains. *Information Sciences*, **418**, 242–257.
276. **Zhang, Z., T. Wei, W. Zhou, H. Zhao, W. Zhang, and N. Yu**, Facersa: Rsa-aware facial identity cryptography framework. *In Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38. 2024.
277. **Zhao, H., W. Zhou, D. Chen, T. Wei, W. Zhang, and N. Yu**, Multi-attentional deep-fake detection. *In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021.
278. **Zhao, L., Q. Liu, L. Yue, W. Chen, L. Chen, R. Sun, and C. Song**, Comi: Correct and mitigate shortcut learning behavior in deep neural networks. *In Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2024.
279. **Zheng, X., Y. Guo, H. Huang, Y. Li, and R. He** (2020a). A survey of deep facial attribute analysis. *International Journal of Computer Vision*, 1–33.
280. **Zheng, X., H. Huang, Y. Guo, B. Wang, and R. He** (2020b). Blan: Bi-directional ladder attentive network for facial attribute prediction. *Pattern Recognition*, **100**, 107155.
281. **Zhong, Y. and W. Deng** (2022). Opom: Customized invisible cloak towards face privacy protection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **45**(3), 3590–3603.

282. **Zhong, Y., J. Sullivan, and H. Li**, Face attribute prediction using off-the-shelf cnn features. *In IEEE International Conference on Biometrics*. 2016a.
283. **Zhong, Y., J. Sullivan, and H. Li**, Leveraging mid-level deep representations for predicting face attributes in the wild. *In IEEE International Conference on Image Processing (ICIP)*. 2016b.
284. **Zhong, Z., L. Zheng, G. Kang, S. Li, and Y. Yang**, Random erasing data augmentation. *In Proceedings of the AAAI conference on artificial intelligence*, volume 34. 2020.
285. **Zhou, P., X. Han, V. I. Morariu, and L. S. Davis**, Two-stream neural networks for tampered face detection. *In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2017.
286. **Zhou, X., K. Jin, Q. Chen, M. Xu, and Y. Shang** (2018). Multiple face tracking and recognition with identity-specific localized metric learning. *PR*, **75**, 41–50.
287. **Zhu, B., H. Fang, Y. Sui, and L. Li**, Deepfakes for medical video de-identification: Privacy protection and diagnostic information preservation. *In ACM Conference on AI, Ethics, and Society*. 2020.
288. **Zi, B., M. Chang, J. Chen, X. Ma, and Y.-G. Jiang**, Wilddeepfake: A challenging real-world dataset for deepfake detection. *In Proceedings of International Conference on Multimedia*. 2020.
289. **Zong, Z., G. Song, and Y. Liu** (2022). Detrs with collaborative hybrid assignments training. *arXiv preprint arXiv:2211.12860*.