VT-NOOD : Vision Transformer based Near OOD Detection in Fine-grained datasets

A Project Report

submitted by

KAUSHIK DEY

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING
INDRAPRASTHA INSTITUTE OF INFORMATION TECHNOLOGY DELHI

NEW DELHI- 110020

August, 2024

THESIS CERTIFICATE

This is to certify that the thesis titled VT-NOOD : Vision Transformer based Near OOD Detection in Fine-grained datasets, submitted by **Kaushik Dey**, to the Indraprastha Institute of Information Technology, Delhi, for the award of the degree of **Master of Technology**, 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.

Dr. Ranjitha Prasad
Thesis Supervisor
Assistant Professor
Dept. of Electronics and Communication
IIIT Delhi, 110020

Place: New Delhi

Date: 4th August 2024

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my advisor Dr. Ranjitha Prasad for her supervision, guidance and constant support. Her expertise in Bayesian Machine Learning has been invaluable in formulating the research methodology. I will be grateful to my advisor for providing me with this opportunity regardless of my lack of research experience. I am deeply grateful and pleased to collaborate with LightMetrics Technologies Pvt. Ltd team for this research project. I would like to thank Vidya T and Ashish Sethi from Lightmetrics team to help to through the projects from formulating the pipelines and writing better code. Lastly, I would like to thank my family for their constant encouragement and motivation throughout this thesis. They stood beside me during my weak days. I also thank my friends with whom I could connect and discuss my problems most of the time.

ABSTRACT

Machine Learning models output high confidence scores when provided with input samples that belong to the input data distribution, referred to as in-distribution (ID) samples. However, when presented with an Out of Distribution (OoD) sample during inference, the same model often outputs uncertain confidence scores raising the question of the interpretability and reliability of these models.

Typically, ML models are incapable of detecting near-OOD samples, which are perceptually similar, but semantically dissimilar samples closely resembling the input distribution with fine-grained variations. We address the issue of fine-grained variations using vision transformers (ViT) and capture the patch-based correlations through the self-attention mechanism. The Vision Transformer is a part of our VAE architecture along with a neural network which helps in patch-level disentanglement. Such a patch-level disentanglement using a ViT encoder results in disentangling the common latent factors for the entire image. To the best of our knowledge, this is the first work that uses a ViT with encoder-decoder architecture for OOD detection. Using experiments on fine-grained datasets such as Oxford Flowers102 and CUB200 Birds Dataset, we demonstrate that the proposed method outperforms OOD-aware baselines in terms of several OOD metrics. Our framework outperforms the density based techniques and classification based methods as compared on OOD Detection metrics such as AUC, AUPR and FPR@95. We also demonstrate that training the entire network with SR-GAN decoder with a combination of Mean Square Loss and Perceptual Loss can learn better representations for density-based methods.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Deep Learning (DL) algorithms have become very successful in a variety of tasks ranging from cancer detection to object segmentation, but it does not lead to reliable predictions. It is known that, often, a deep learning model trained to classify between dogs and cats gives higher probability scores for inputs like a car or a building which is vastly dissimilar to the in-distribution (ID) sample Yang *et al.* (2024). Such inputs are generally termed as out of distribution (OOD) samples and identifying these inputs during inference is crucial for DL models deployed in safety critical applications Wang *et al.* (2024).

Despite the excellent capabilities illustrated by the classical supervised learning framework, it cannot be directly deployed to deal with the OOD generalization problem. One of the most fundamental assumptions of the classical supervised Learning framework is the independent and identically distributed (IID) assumption for training and testing samples. Due to this assumption, the model which is trained by absorbing the correlations found in the ID data through minimizing the loss performs poorly when exposed to unseen testing distributions where the majority of correlations do not hold. These unseen samples can look perceptually similar to the training distributions having completely different factors of variation for the important parts of the image (semantically dissimilar) Mukhoti *et al.* (2023). Samples from such similar-looking distributions are often categorized as Near OOD samples. Our work focuses on fine-grained classification with an increased robustness to OOD samples.

1.1 OOD samples and OOD Detection methods

Among several sources of OOD samples, common causes include dataset shift , training and test input follow different distributions, dataset shift can occur due to several issues like sampling bias. However, concept shift can occur mostly in time series problems

where the relationship between features and labels changes in due time Zhang *et al.* (2024).

Several popular approaches have emerged towards OOD detection. In Yang *et al.* (2024), the authors survey several methods for Out of Distribution Detection like classification based, density-based, distance-based and reconstruction based. Classification-based methods generally use the confidence scores to metrics like MSP Hendrycks and Gimpel (2018), LogitNorm Wei *et al.* (2022) and several other metrics for detecting OOD samples during inference. However, these metrics tend to be biased to input distribution as the majority of the works never train the model with OOD samples. Further, we have density-based and reconstruction-based methods which use the reconstructed image or the latent space of learned representations to detect OOD samples Xiao *et al.* (2020). The separation of learned distributions taking the score as the random variable further helps to detect OOD samples.

Apart from these, distance-based metrics like Mahalanobis distance Denouden *et al.* (2018) finds the distance between the representation of learned latent space to the input sample during test time. To determine whether the sample is from the training distribution or OoD several metrics have been proposed. In Xiao *et al.* (2020), the authors propose a likelihood based metric in which the encoder network is retrained to learn the disentanglement of the input images during inference. Another such metric in that direction is Ren *et al.* (2019) where the authors have corrupted the semantic parts of the image stating that background statistics influence the OOD detection. Popular metrics also include AUROC Tafvizi *et al.* (2022), AUPR and FNR95.

1.2 Disentangled Representation Learning

Disentangled Representation Learning (DRL) aims to learn models that can learn to disentangle meaningful factors or learn the factors of variations in any given dataset. Popularly, disentangled representation is defined as:

Definition 1: Disentangled representation should separate the image into single distinct, independent, and informative generative factors of variation in the data. These single latent variables are sensitive to changes in underlying generative factors while being relatively invariant to changes in other factors.

We classify DRL using the base-level architectures that they use for learning the low-level representations. Among other techniques, Variational auto-encoder (VAE)-based and GAN-based techniques are popular Chen *et al.* (2016). VAE comprises of encoder and decoder architecture where the role of the encoder is to learn the disentangled representation whereas the role of the decoder is to reconstruct the original image from the low dimensional disentangled latent vector . One of the most important characteristics of using VAE-based DRL is the benefit of dimension-wise structure where each dimension represents different factors of variation. GAN based methods directly samples latent representations by training process where the generator network takes a noisy input and latent vector as input and reconstruct it to a fake image perceptually similar to the input image but from a different distribution.. The discriminator plays and important role by classifying whether the data is real or fake Chen *et al.* (2016). In this work, we employ a VAE based architecture to do patch level disentanglement. We show that patch level disentanglement proves to give better disentanglement than the traditional dimension wise and vector wise disentanglement.

1.3 ViT and OOD

ViT models have proved to be more powerful than the classical convolutional neural network due to their ability to use the attention mechanism. ViT has been extensively used for object detection, image classification and action recognition. ViT models comprises of multiple self attention or multi head attention as a part of the encoder. The self attention mechanism does a patch to patch based correlation by learning the weight of the self attention layer. These correlations contribute to the attention map by giving high importance to the patch for a particular class .

Recent works have shown that the attention map is capable of capturing the intra class relations . Attention maps have been used to detect out of distribution samples during test time using confidence scores. Koner *et al.* (2021). The disentangled representations of the attention Maps have also been used to detect out-of-distribution samples Cultrera *et al.* (2023). Therefore we can say that the attention maps are well capable to capture Near OOD and Far OOD samples during the test time. In this work, we use a VIT to achieve patch-wise disetanglement for OOD Detection.

1.4 Contributions

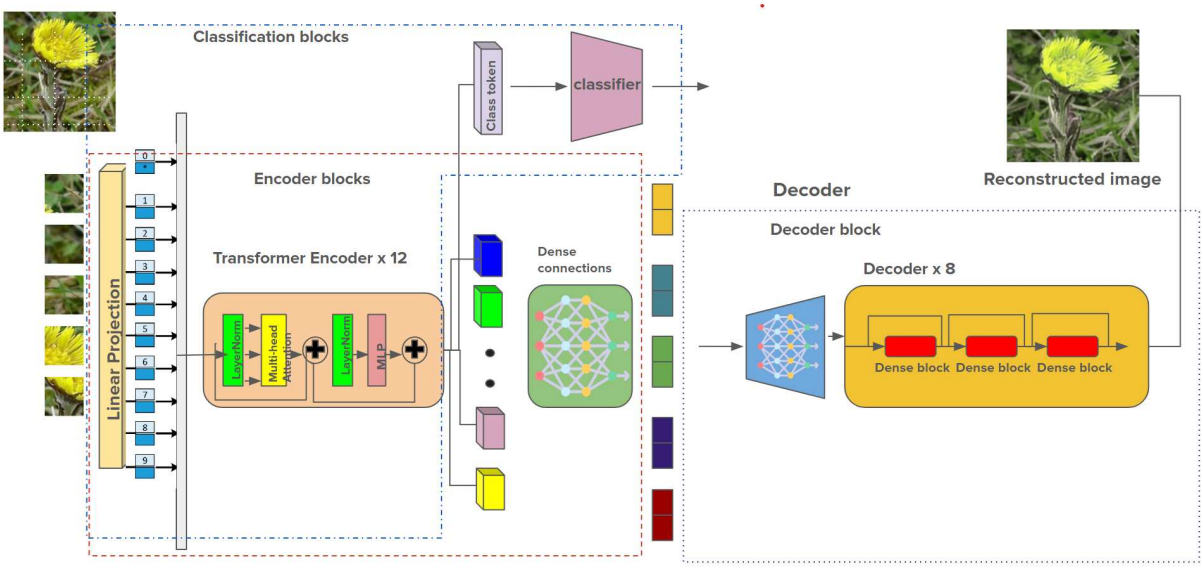


Figure 1.1: Training pipeline of our work with custom designed decoder

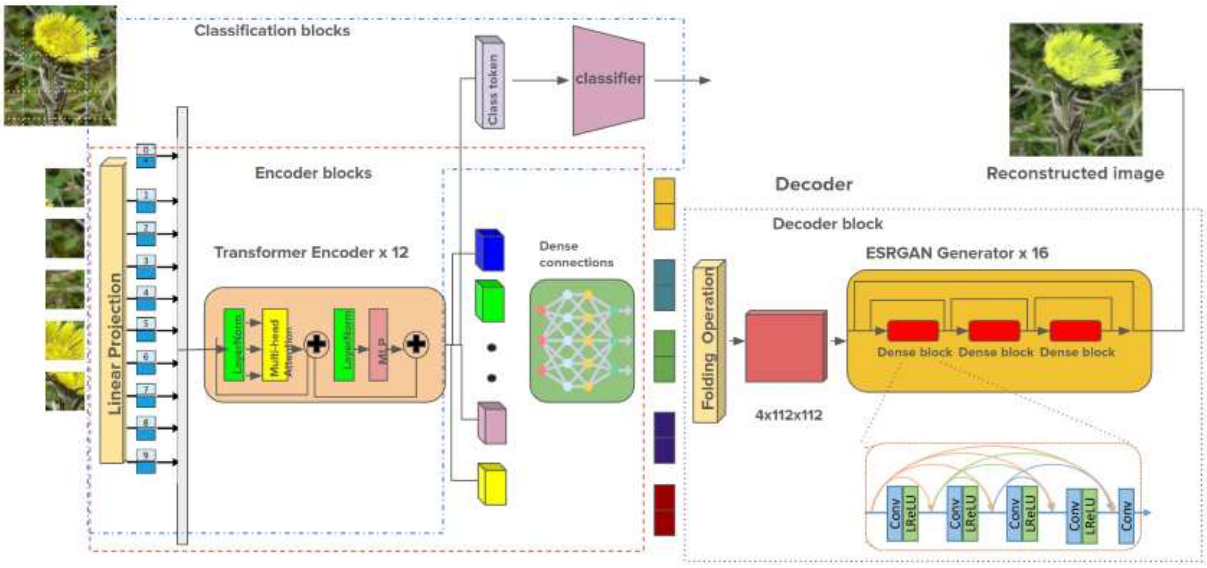


Figure 1.2: Training pipeline of our work with ESRGAN decoder

In this work, we propose Vision Transformer based Near-OOD detector (VT-NOOD) for detecting far and near OoD samples during inference. We show that our work outperforms the existing density-based methods such as Likelihood Regret on metrics like AUROC, AUPR, FPR@95. The architecture consists of following parts:

- **Reconstruction branch:** This branch is similar to similar to a variational autoencoder, and consists of a encoder and a decoder. The encoder is a ViT along with a neural network which is responsible for disentangling the feature representation per-patch of the image sample. We use super-resolution GAN as the decoder for image reconstruction.

- **Classification Branch:** This branch consists of a ViT along with a classifier network which is a dense neural network. The classifier utilizes the combination of patch-level feature information and attention weights for efficient classification.

1.4.1 Notations

Small letters denote scalars, boldface small letters denote vectors. $\mathbf{I}$ denotes an identity matrix whose size is as per context. The ℓ_2-norm of a vector $\mathbf{x}$ is denoted as $||\mathbf{x}||$. $\mathcal{P}$ represents sets and $|\mathcal{P}|$ represents size of the set.

CHAPTER 2

RELATED WORKS

Out-of-distribution detection has been an important research topic in the domain of uncertainty quantification. This chapter gives an overview on the works that have been carried out in this direction. We propose a novel VT-NOOD framework for near-OOD detection, which is related to some of the methods presented in this chapter.

2.1 Near OOD Detection

Several works have been carried out to detect near OOD samples. The basic definition of Mukhoti *et al.* (2023) proposes the fundamental difference between ID and other types of OOD samples. They propose that near OOD samples are perceptually similar but semantically different, whereas far OOD samples are both perceptually dissimilar and semantically different.

Another similar work Singla *et al.* (2023) shows how augmenting from a counterfactual explainer lessens the uncertainty on the confidence scores of the model. Their work actually confirms the definition proposed by Mukhoti *et al.* (2023) as the counterfactual images are often very similar looking but they have causal alternations. Some methods have emphasized that the penultimate layer of the deep neural networks can capture the class level variations and therefore applying distance based methods can separate the near OoDs. In Sun *et al.* (2022), the authors demonstrate that contrastive learning with K-Nearest Neighbours can separate Near OoD clusters from ID. In Sun *et al.* (2021), the authors showed through experimental results that the penultimate layer of a neural network shows different patterns when the network is exposed to an OoD sample. They propose a method ReAct which suppresses overconfident classifiers.

2.2 Vision Transformer Based Near OOD Detection

Attention Maps are often quite useful in explaining the prediction of the model. In recent years, few works have tried to explore the attention maps of transformer networks to detect OOD samples Abnar and Zuidema (2020). In Cultrera *et al.* (2023), the authors use attention maps to detect near OOD samples in wildlife images. As the attention heatmap focuses on the important parts while detecting an object belonging to a specific class, the latent vector of the heatmap capture semantic aspects. They show that reconstructing the attention heatmap using convolutional autoencoder captures the details of ID samples and the reconstructed image is shown to be sufficient to detect OOD samples. Another work in the same direction Sim *et al.* (2023) emphasizes that less attended parts of the image are responsible for Logit based metrics to under perform. They change the architecture of Vision Transformer to mask the unattended features and thereby show that the metrics like Mahalanobis distance improves. In Koner *et al.* (2021), authors use the Vision transformer as the feature extractor to exploit the object concepts and discriminatory attributes along with co-occurence via visual attention. They propose a new detection algorithm which uses class token which is a combination of global attention and features from previous blocks. They calculated distance between the mean vectors of ID and OOD class tokens.

Near OoD detection based on fine grained classification datasets is a challenging task for most of the deep learning frameworks due to subtle differences between classes, however, Fort *et al.* (2021) explores how vision transformer is powerful than other methods in detecting near OoD samples using classification based methods like MSP Hendrycks and Gimpel (2018) and Mahalanobis distance Denouden *et al.* (2018). This method uses few shot outlier exposure setting where few samples are available to the classifier during training. Delving deeper into the ViT's Zhang *et al.* (2022) observed that ViT's weakly respond to backgrounds and textures while they respond strongly to shapes and structures which are more consistent and human like traits. This helps ViT's stand out from the traditional Convolutional Neural Networks in OoD detection.

2.3 How disentanglement helps in OoD Detection

Multiple works have focussed to improve the learned representation during the training stage so as to achieve a better OoD detection during inference. Contrastive representation Learning targets learning a discriminative representation space while positive samples align while negative samples are removed, this shows to perform well in OoD Detection Zhang *et al.* (2023).

Some works suggest that reconstruction-based out-of-distribution methods perform better in detecting out-of-distribution samples. They suggest that the reconstruction becomes poorer when a learned model receives an out-of-distribution sample. Jiang *et al.* (2023) proposes a new method for autoencoder training such that the losses converge for in-distribution samples and the reconstructed images can be used to detect out-of-distribution in real-time. Yang *et al.* (2022) shows how masking some important parts and forcing the generator to generate only a small portion of the entire image increases the capabilities of the model to detect out-of-distribution samples through reconstruction. Denouden *et al.* (2018) have tried to detect out-of-distribution samples by finding the distance between a distribution learnt during VAE training and the input samples. Mahalanobis distance is used as a distance based metric to detect out of distribution during training.

2.4 Novelty

In this work, we propose a VT-NOOD framework that can detect Near OOD and Far OOD samples. Our framework is a combination of a classifier and an encoder-decoder type VAE architecture. We have used the vision transformer as our backbone encoder along with a neural architecture for patch-level disentanglement. Generally, dimension wise DRL cannot capture high level factors of variation as vector wise DRL. However, we show that by disentangling the patches with a shared encoder the patch level alterations can be captured. We observe that when a latent factor is changed it effects the entire image but some part of the image may not respond to that latent factor.

The encoder-decoder part of our network ensures OOD Detection through the density-based methods. We show that using patch level disentanglement we achieve almost

identical reconstruction using ESRGAN generator network as the decoder which guarantees that our work is inline with the vector wise DRL method. Also, we show that our framework outperforms the other baseline models in AUROC, AUPR, FPR@95 metrics for both Near and Far OOD samples. We also show that our work using a custom decoder in the architecture performs better than the baselines using the classification-based OoD detection methods when compared along the same metrics.

CHAPTER 3

Preliminaries and the Proposed Method

In this chapter, we first introduce the fundamentals of dimension-wise DRL, VITs and SR-GAN decoder used in our framework. Later, we proceed to describe the proposed method, namely the VT-NOOD technique for near-OOD detection using VITs.

3.1 Disentanglement Representation Learning

Disentangled representation learning aims to break down complex data into its underlying independent factors of variation. In other words, its about learning a representation where each dimension of the representation captures a distinct aspect of the data.

According to Wang *et al.* (2024) representational structure of DRL can be broadly categorized into the following :

- **Dimension-wise DRL**: The main aim of such a structure to learn a representation **z** where each dimension in the representation learns one fine grained generative factor. Dimension-wise DRL is often used with simple and synthetic datasets like Lecun *et al.* (1998) which have fine grained latent factor which cannot be learnt in a real world and complex datasets like Wah.
- **Vector-wise DRL**: This type of representation aims to learn a set of course grained representations where each representation vector **z** is a dimension wise disentanglement of a factor of variation. Vector-wise DRL is often used with real world complex datasets like Liu *et al.* (2021) where they used vector wise disentanglement to separate video representation to appearance and motion parts.

The fundamental difference between dimension-wise DRL and vector-wise DRL is their ability to capture the amount of information or in other words their information capacity. Dimension-wise disentanglement tends to be less informative than vector-wise disentanglement.

To disentangle the factors of variations and to learn a representation vector researchers commonly adopt two types of backbone architectures

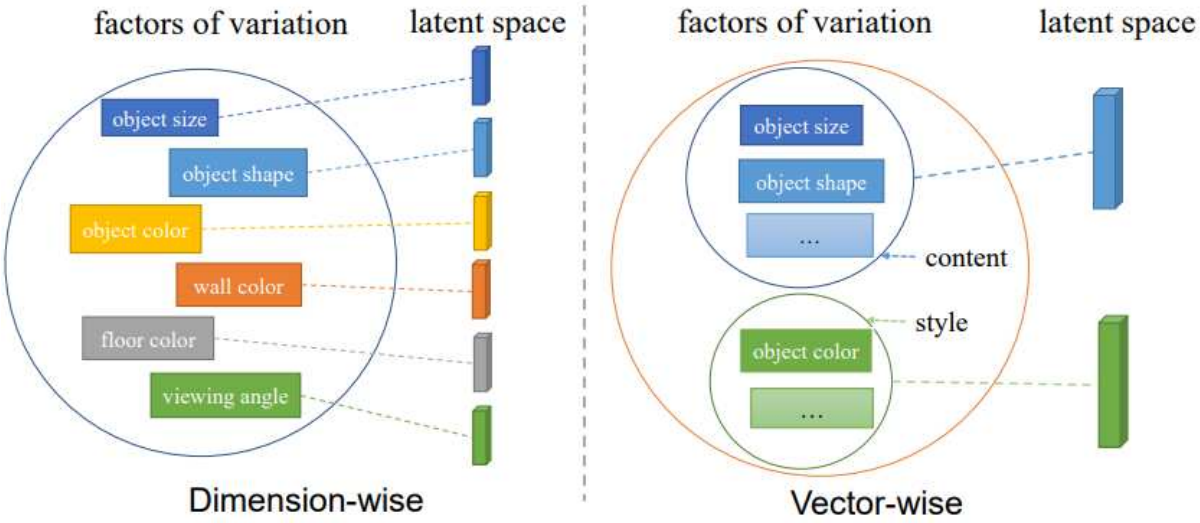


Figure 3.1: Comparison between dimension wise and vector wise disentanglement

VAE based DRL methods Variational AutoEncoders(VAEs) have shown to be of great importance for learning disentangled representations. The fundamental idea of VAE is to model data distributions using variational inference. The objective of VAEs is to minimize the ELBO loss which is defined in the equation below:

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}) = -\mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \tag{3.1}$$

Generally a standard gaussian distribution prior $N(0,\mathrm{I})$ is chosen for $p_\theta(z)$. While training the VAEs initially sample from the prior but slowly learns the underlying latent space. The KL Divergence term imposes independent constraints on the latent factors and hence we can assure that VAE does disentanglement.

Several works have been carried out to improve the performance of VAEs like β-VAE Higgins *et al.* (2017), FactorVAE Kim and Mnih (2019). Generally, VAEs compress information through the bottleneck layer during disentanglement leading to information loss and poor reconstruction. VAEs often face a tradeoff between learning good representations and reconstructing identical images. Therefore β-VAE solves this problem by introducing a hyperparameter β in the ELBO loss which gives priority to each of the sub-tasks as follows:

$$\mathcal{L}(\theta, \phi; \mathbf{x}, \mathbf{z}) = -\beta\mathcal{D}_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]. \tag{3.2}$$

GAN based DRL methods

Generative adversarial networks (GAN) Goodfellow *et al.* (2014) is another important generative model which can be used for DRL task. Instead of using statistical methods GANs directly sample from the prior distribution $p_\theta(z)$. Generally, GANs have a generator and discriminator network. The generator network directly samples from the prior distribution and transforms it into an image, whereas the discriminator network classifies the generated image to be real or fake. After the training procedure the generator network learns to generate fake images which come from different distribution but look similar to the training data. The training loss of the GANs is given below.

INFOGAN Chen *et al.* (2016) being the earliest work to find a disentangled representation of data using a GAN based approach. The INFOGAN generator takes both the noisy image output and noisy latent representation and generates the image. The authors introduce an auxiliary network which forces the generator to generate images which belong to the latent space.

In the context of VITs, we are able to obtain patch-level disentanglement. Before introducing the proposed technique, we describe the fundamentals of VITs in the following section.

3.2 Preliminaries on VIT

Transformer was first introduced in the paper "Attention is All You Need" Vaswani *et al.* (2023) which showed the power of transformers for natural language processing tasks. Although transformers have been extensively used for natural language processing tasks, Dosovitskiy *et al.* (2020) gave us an alternative to convolutional neural networks by introducing the concept of ViT. ViTs have become popular due to their ability to use the self-attention mechanism on different patches of an input image highlighting the fact that understanding the relationship between different parts of an image contributes effectively to the downstream tasks such as classification. Vision Transformers are very good at explainability as the attention weights calculated through self-attention layers help in identifying the region of focus during predictive tasks, hence increasing the reliability of predictions.

Self Attention: This is the basic form of attention that calculates the relative weight of each patch of an image with the others. Firstly, the input of self-attention which are flattened patches with positional encodings is used to find three vectors Query, Key, and Value. These three vectors are computed by learning the weights corresponding to these vectors using namely $\mathbf{W}_q \in \mathbb{R}^{P\times d}$, $\mathbf{W}_k$, $\mathbf{W}_v$. The optimal Query, Key, and value vectors thus obtained during training are used to calculate the self-attention weights using the equation below:

$$\frac{softmax(\mathbf{Q} * \mathbf{K}^T)}{\sqrt{d_k}} * \mathbf{V}, \tag{3.3}$$

where $*$ indicates matrix multiplication.

Self attention captures the inter correlation of given input matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots] \in \mathbb{R}^{n\times d}$. In the self attention, we let $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}$. Therefore the equation becomes.

$$\mathbf{O} = \mathbf{W}_v\mathbf{X} * softmax((\mathbf{W}_k\mathbf{K})^T\mathbf{W}_q\mathbf{Q}) \tag{3.4}$$

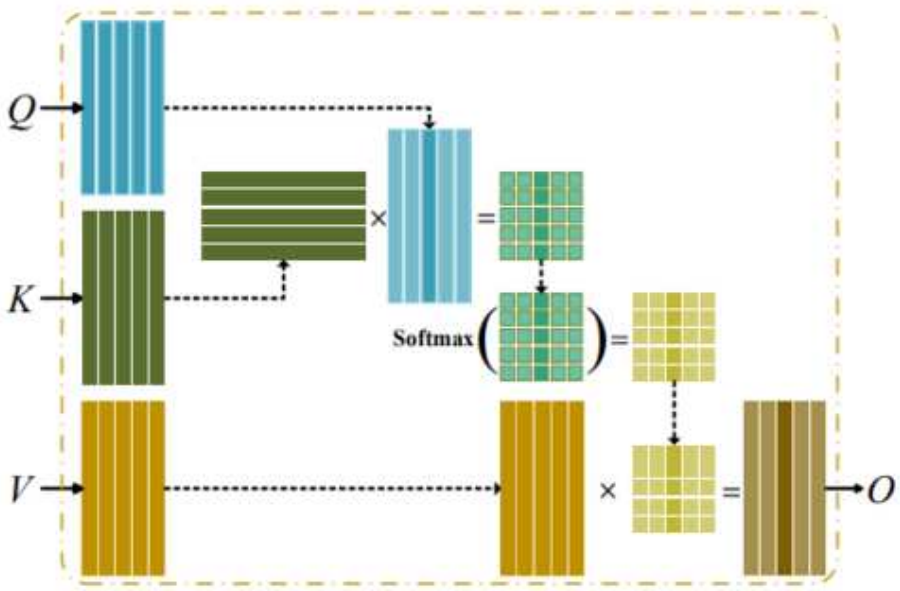


Figure 3.2: Mechanism of attention and Self Attention

Multi Head Attention: It behaves as multiple readers analyzing data from different angles. It splits the input information into queries, keys, and values and then calculates the attention scores to weigh which values are most relevant to each query based on the keys. These weighted values from multiple heads are then combined to create a richer and more nuanced understanding of the data. This allows ViTs to capture not just local

features but also complex relationships across different parts of an image, leading to impressive performance in computer vision tasks.

3.3 ESRGAN Decoder

The enhanced SRGAN (ESRGAN) generator network is the heart of the ESRGAN process, responsible for taking a low-resolution image and transforming it into a high-resolution version. The key components are as follows:

Building Block: Residual-in-Residual Dense Block (RRDB)

Unlike standard convolutional neural networks (CNNs), ESRGAN utilizes a special building block called the Residual-in-Residual Dense Block (RRDB). This block consists of multiple dense convolutional layers stacked together. Dense connections mean each layer receives input from all preceding layers, promoting better information flow and feature reuse. The "Residual-in-Residual" part involves adding the input of each RRDB block to its output, allowing the network to learn residual functions and focus on capturing the additional details needed for high-resolution reconstruction. Leveraging Multiple RRDBs:

The ESRGAN generator network stacks several RRDB blocks one after another. This creates a deep architecture capable of progressively refining the image and capturing increasingly intricate details. Channel Growth and Feature Extraction:

As the network progresses through the RRDBs, the number of channels typically increases. This allows the network to extract and represent more complex features necessary for high-resolution reconstruction. Upsampling Strategy:

To achieve the final high-resolution output, ESRGAN employs a specific upsampling strategy within the generator network. This strategy can involve techniques like pixel shuffling or transposed convolutions. Pixel shuffling rearranges the data from the previous layer to create a higher-resolution feature map. Transposed convolutions essentially learn to "up-sample" the feature maps, increasing their resolution. Overall, the ESRGAN generator network leverages residual connections, dense convolutional layers, and a strategic upsampling approach to effectively transform low-resolution images into high-resolution counterparts with enhanced detail and visual quality.

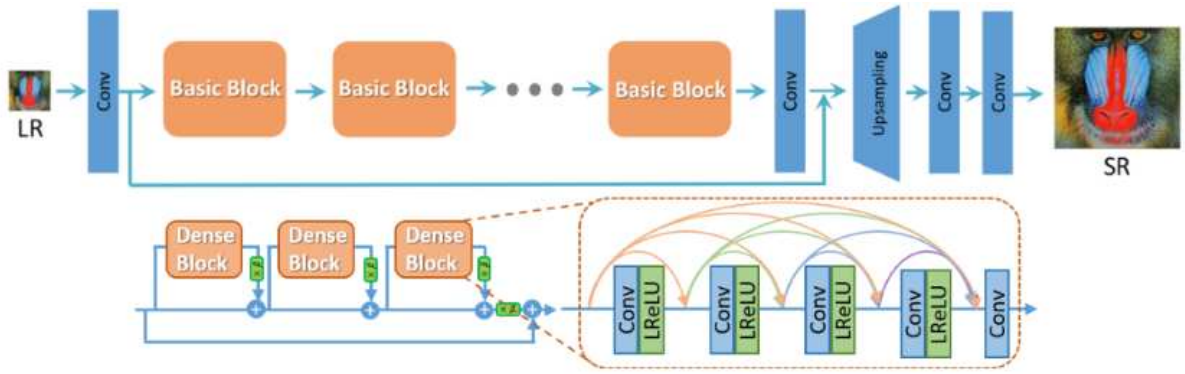


Figure 3.3: Architecture of Enhanced Super-resolution GAN

3.4 Proposed Framework: VT-NOOD

This section describes our pipeline along with the training procedure. As shown in Fig. **??**, the proposed pipeline consists of three components namely, the VIT based encoder, decoder, and the classifier. Both the classifier and decoder are dependent on the encoder for the specific task. The encoder is a combination of two neural network models, firstly the vision transformer encoder which outputs the patch-level features, and additionally a feed-forward neural network model which takes the features and disentangles them into low dimensional representations. The feed-forward neural network is a shared network for all the patches and hence, patch level change in representations can be captured using this framework. The entire combination acts as an encoder for our pipeline.

Along with patch-level features, we have a class token that contains class-level information about patches. The input to the classifier is the class token and the output is the predicted class. In the forward pass, the class token passes through the mutihead-attention units and hence has the information of all the features and their attention weights, globally. As the patch-level features are well-trained for disentanglement the classifier gets trained to predict only the ID samples with minimal uncertainty.

After the patch-wise fine-grained features are passed through the feed-forward neural network, the latent vectors of each patch are folded to form an image with a lower spatial size than the original one. This folded image then is passed into the decoder which is an SRGAN generator network. The decoder has several RRDB blocks as discussed in the previous sections. In case of our custom decoder we do not use any folding operation but a dense network is added to reconstruct the latent space back to the patch size.

3.4.1 Proposed Decoder

VAE architecture consists of an encoder for encoding or disentangling the input image into a latent vector which further reconstructs the latent representations back a perceptually similar output image. To complete our proposed pipeline we introduce our custom designed decoder which consists of 12 decoder blocks each decoder blocks have two deep neural networks and internally follows an architecture similar to residual blocks. Unlike the ESRGAN Decoder our proposed architecture does not require any folding operation to be executed before decoding.However, a dense neural network which is closely the inverse of the encoder network is added to follow the VAE architecture. This architecture closely resembles the backbone ViT encoder architecture and does patch-level reconstruction **??**.

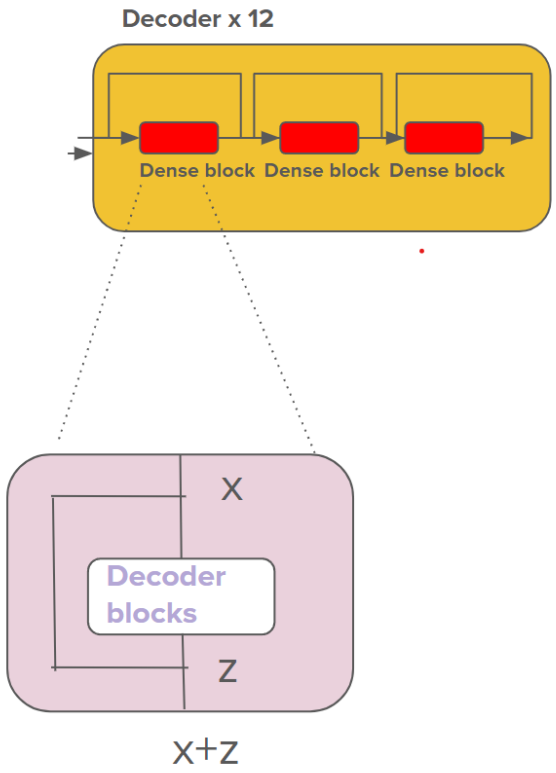


Figure 3.4: Architecture of Our proposed decoder with residual blocks

3.4.2 Proposed Loss Function

Generative models like variational autoencoders often generate poor-quality images due to the bottleneck layer which suppresses the finer details . Using a pixel-to-pixel-based comparison loss like Mean Squared Error (MSE) which depends on low-level infor-

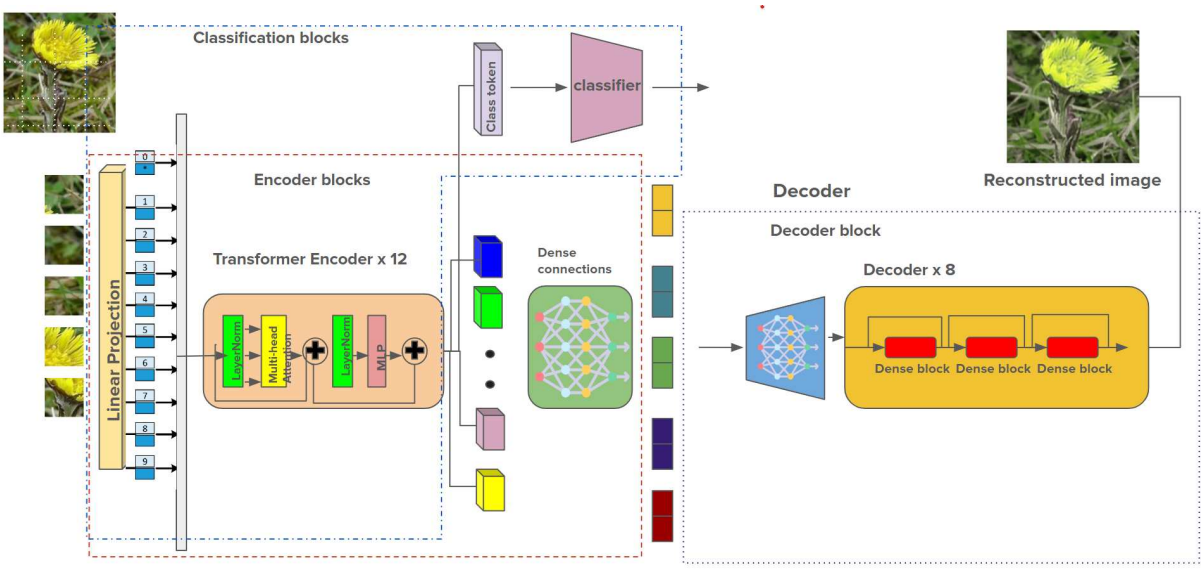


Figure 3.5: Training Pipeline with our own proposed Decoder

mation to learn the weights of the VAE. Instead of MSE, the authors in Johnson *et al.* (2016) propose a new loss function that can capture the style-based features and focuses more on learning layer-wise representations. It penalizes the generator network to output images which look similar to the input image. The input image and the generated images are both passed to the pre-trained frozen network (VGG16) and the change in intermediate feature maps from output activations is used to train the generator network. This loss is popularly known as perceptual loss.

The entire pipeline is trained end to end using two loss functions. The classification task is trained with Categorical Cross Entropy while the Variational architecture is trained using KL divergence and Reconstruction Loss. We have used a combination of MSE and perceptual Loss for the reconstruction. The MSE is used to ensure that the model closely outputs the same pixels as the input image and the perceptual Loss ensures the image style features are intact. Perceptual Loss is given higher weightage to learn the style and artistic features. The expressions for the losses are as follows:

$$\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2, \tag{3.5}$$

where N represents the total number of samples.

Feature Reconstruction Loss: Perceptual Loss is a combination of feature reconstruction loss and style reconstruction loss we achieve our results with only feature

reconstruction loss.

$$\mathcal{L}_{perceptual} = \frac{1}{C_j H_j W_j} \|\phi_j(\hat{y}) - \phi_j(y)\|_2^2 \tag{3.6}$$

Cross Entropy Loss: The classifier branch of our network is trained with categorical cross entropy loss.

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{C} y_{n,i} \log(\hat{y}_{n,i}), \tag{3.7}$$

where C represents the number of distinct classes in the given dataset. Hence, the total loss is given by

$$\mathcal{L}_{proposed} = \mathcal{L}_{MSE} + \kappa \mathcal{L}_{perceptual} + \gamma \mathcal{L}_{CE}, \tag{3.8}$$

where κ and γ are hyperparameters.

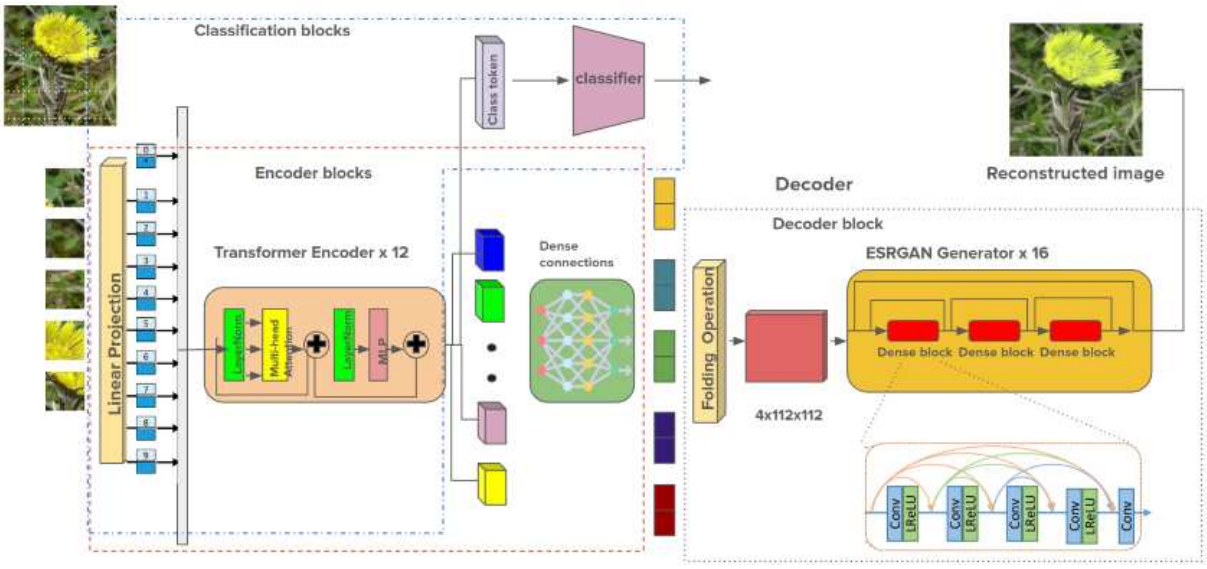


Figure 3.6: Architecture of Enhanced Super-resolution GAN Pipeline

CHAPTER 4

EXPERIMENTS AND RESULTS

In this chapter, we will discuss the experimental setup and also the evaluation metrics along with the datasets we use to train our model. These experiments aim to improve the separation between ID, near OOD distribution, and far OOD distribution. In particular, our experiments showcase the following (a) likelihood regret based performance analysis of the encoder-decoder module (b) Classification based metrics for performance analysis of the classification module.

4.1 Datasets

To compare the efficacy of our architecture with the existing models we use two fine grained datasets.

4.1.1 Oxford Flower Dataset

The dataset comprises of 102 different classes of flowers, commonly found in North America. Each category has between 40 and 258 images, with a total of over 8000 images in the dataset. The images showcase variations in size, pose, and lighting conditions to reflect real-world scenarios. The image size is 224×224.

Near OoD Data The entire dataset is divided into training, testing, and Near OoD Datasets. Out of 102 classes, 80 classes are taken for ID, i.e., training and testing datasets. Rest of the classes are considered to be near OoD samples. In each of the 80 classes, 50 samples constitutes the training data and rest are reserved as testing samples.

4.1.2 CUB200 Dataset

The dataset comprises 200 different classes of Birds. Each category has between 30 to 60 images, with a total of over 11,768 images in the dataset. Fine-grained classification

(a) Desert Rose

(b) Stemless Gentian

Figure 4.1: The above images shows ID and Near OOD samples of Oxford Dataset 102 dataset

(a) Bay Breasted Warbler

(b) Wilson Warbler

Figure 4.2: The above images shows ID and Near OOD samples of CUB200 dataset

tasks like CUB200 are inherently challenging due to the subtle differences between bird species.

Near OoD Data The entire dataset is divided into training, testing, and Near OoD Datasets. Out of 200 classes, 160 classes are taken for In distribution, i.e., training and testing datasets. The rest of the classes are completely considered to be Near OoD samples. Out of the 160 classes from each of the class 50 samples constitute the training data and the rest are reserved as testing samples.

4.2 Augmentations

Due to the limited samples in most of the fine grained datasets, we performed augmentations. We used four different types of augmentations for both the datasets namely,

horizontal flip, vertical flip, random flip with degree = 60, Gaussian blur with kernel of size 3.

4.3 Implementation Details

We have used Nvidia L40 GPU to train our entire framework. We develop the models using Pytorch. The proposed loss in (3.8) is used where the perceptual loss is given more weightage while training, i.e., $\kappa = 10$. We have used two optimizers both being Adam with weight decay with initial learning rate = $1e-5$ with weight decay $1e-2$. First, the optimizer updates the parameters of the VAE while the second optimizer updates the parameters of the classifier. The vision transformer weights are being updated by both the optimizers. We introduce two schedulers for both the losses, the first scheduler being the LRonPlateau which steps down the learning rate by a factor of $1e-2$ if no improvements are observed in the reconstruction for 1 epoch, while the other scheduler is an exponential decay with $\gamma = 0.9$ for every 3 epochs. For custom decoder we modify the weight of classification task to $1e-3$.

4.4 Baselines

We have used the baselines listed below for the comparison and evaluation of framework:

- Likelihood-based metrics have been used to detect OOD samples such as Likelihood regret Xiao *et al.* (2020) for a VAE based architectures.
- Attention Mechanisms have been used to detect OoD samples for both far OoD and Near OoD samples. Sim *et al.* (2023) showed OoD detection using attention masking.
- Cultrera *et al.* (2023) closely resembles our work but uses attention maps to train autoencoder. Compares various metrics like AUROC, AUPR with similar architecture.

4.5 Performance Metrics

In this section, we list the performance metrics used to demonstrate the near-OoD detection efficacy of the encoder-decoder as well as the classifier.

4.5.1 Near-OoD detection Metrics for Encoder-decoder

Likelihood Regret: Likelihood Regret (LR) measures the log-likelihood improvement of the model for a sample during test time. Generally, generative networks try to learn a input distribution during training time. VAEs learn the latent space from the input data. However, these model are input data dependent and changes in the input distribution does effect these models. LR measures the change in the latent space by the input sample during test time. It measures the difference between the log-likelihood of the change in the latent space.

$$\mathcal{L}_K(x;\theta,\phi) = \log p_\theta(x) \geq \mathbb{E}_{[z^1,z^2,z^3\ldots z^k]\ q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(\mathbf{x}|z^k)p(z^k)}{q_\phi(z^k|\mathbf{x})}\right]. \quad (4.1)$$

We represent the difference in likelihood with optimal parameters as $\mathcal{L}_K(x;\theta^*,\phi^*)$.

Algorithm 1 Computing Likelihood regret

Require: Test sample **x**, trained VAE $(E_{\phi^*}, D_{\theta^*})$, number of posterior samples for likelihood estimation , number of optimization step s, learning rate l

$L_{VAE} = \mathcal{L}_K(x;\theta^*,\phi^*)$ ▷ Estimate the log likelihood with VAE model

Set $\phi = \phi^*$

for S iterations **do**:

$\phi \leftarrow Adam(\phi, \nabla_\theta(-\mathcal{L}_K(x;\theta^*,\phi^*)), l)$

end for

$L_{OPT} = \mathcal{L}_K(x;\theta^*,\phi^*)$

$LR = L_{OPT} - L_{VAE}$

4.5.2 Near-OoD detection Metrics for Classifier

We also utilize classifier specific OoD metrics as follows:

Entropy: Entropy is a measure of uncertainty or randomness which in the context of out of distribution detection is applied to the softmax outputs.

Generally ID data samples are more confident and often leads to low entropy while OOD samples are completely new to the models and therefore the classifier assigns high entropy values. Entropy measure can assign higher entropy values to even ambiguous ID samples and hence using it to detect near OOD samples becomes easier. Some works have also explore multiple methods to optimize the entropy Macêdo *et al.* (2021) by introducing IsoMax loss as an alternative.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{4.2}$$

MSP: The maximum softmax probability in the output of a softmax function tells you the predicted class and its associated confidence for a multi-class classification problem.

Softmax Function: This function takes a vector of real numbers as input (often from the output of a neural network) and converts them into a probability distribution over a set of possible classes.

Output Probabilities: The softmax function outputs a vector of values between 0 and 1, where each value represents the probability of the input belonging to a specific class. Maximum Probability: The maximum value in this output vector is the maximum softmax probability.

4.6 Evaluation Metrics

In this section we briefly descibe about the three evaluation metrics we used to make an unbiased comparison with other related works.

- **AUROC** : AUROC stands for Area Under the Receiver Operating Characteristic Curve. It's a metric used to evaluate the performance of classification models, especially binary classifiers. AUROC of 1 means a perfect classifier and AUROC of 0.5 means random guess. AUROC is a metric which is invariant to class imbalance and considers all possible classification thresholds giving a complete picture of the model performance.

- **AUPR** : AUPR stands for Area Under the Precision-Recall Curve. It's a metric used to evaluate the performance of classification models, particularly in scenarios with imbalanced datasets. It focuses more on the positive classes which which is the class of interest.
- **FPR@95** : FPR@95 is a metric used in classification, particularly in the context of imbalanced datasets and anomaly detection.

  It represents the False Positive Rate (FPR) when the True Positive Rate (TPR) is at 95% . A lower FPR@95 indicates better performance, as it means fewer false positives while still achieving a high true positive rate.

4.7 Experimental Results

In this section, we first compare the performance of the proposed scheme with the above mentioned baselines. We later perform ablation study to understand the behavior of the proposed technique.

4.7.1 Comparison with Baselines

We have evaluated our work with the baselines mentioned above. Most of the baselines have used pretrained ViT in their pipelines for classification task and also for classification based out of distribution detection methods.

The evaluation of various OOD Detection methods for both the mentioned datasets is reported in 4.1 while the same set of evaluations is carried out for far OOD Detection and reported in 4.2.

From the table the following points can be inferred:

- We observe that for Near OOD Detection our method outperforms the baseline models for classification based metrics and density based methods.
- VT-NOOD also does well with density-based methods like Likelihood Regret and Likelihood Ratios.

4.7.2 Comparison of Proposed Decoder with ESRGAN Decoder

We have trained our proposed pipeline with an ESRGAN decoder and a custom decoder. While the classifier performs poorly in the SRGAN Decoder, the reconstruction outputs

Table 4.1: Comparison of various scores using AUROC, AUPR, FPR@95 for Near OOD Detection using Vision Transformer as the baseline

Dataset	Model	Method	AUROC	AUPR	FPR@95
Flowers102	Baseline	LR	**0.527**	0.516	0.98
		MSP	0.6910	**0.3782**	0.7414
		Entropy	0.3129	0.7136	0.98
	Ours	LR	0.4268	**0.6229**	**0.8811**
		MSP	**0.9529**	0.3107	**0.3373**
		Entropy	**0.9561**	**0.9357**	**0.1449**
CUB200	Baseline	LR	0.4411	**0.6175**	**0.8613**
		MSP	0.8332	**0.3342**	0.7
		Entropy	0.8585	0.8329	0.4495
	Ours	LR	**0.4985**	0.4806	0.97
		MSP	**0.8657**	0.3268	**0.618**
		Entropy	**0.884**	**0.8644**	**0.408**

Table 4.2: Comparison of various scores using AUROC, AUPR, FPR@95 for Far OOD Detection using Vision Transformer as the baseline

Dataset	Model	Method	AUROC	AUPR	FPR@95
Flowers102	Baseline	LR	**0.645**	0.617	**0.861**
		MSP	**0.9991**	0.3236	**0.0025**
		Entropy	**0.9991**	**0.9994**	**0.0**
	Ours	LR	0.3113	**0.7524**	0.9603
		MSP	0.9837	**0.3254**	0.0935
		Entropy	0.987	0.9821	0.062
CUB200	Baseline	LR	**0.8815**	**0.7325**	**0.4752**
		MSP	0.9443	**0.3126**	0.2865
		Entropy	0.9679	**0.9695**	**0.1485**
	Ours	LR	0.4983	0.4909	0.9405
		MSP	**0.9595**	0.3102	**0.195**
		Entropy	**0.9684**	0.9675	0.181

are very sharp. We tried our density-based metrics with the SRGAN decoder and found that our results beat the existing work by a huge margin. We show some of our results in this section.

4.7.3 Low Dimensional Visualization of class token

We provide the plots for the entropy and MSP of the proposed technique on all the datasets. We observe from the entropy plot that while the ID sample leads to a low

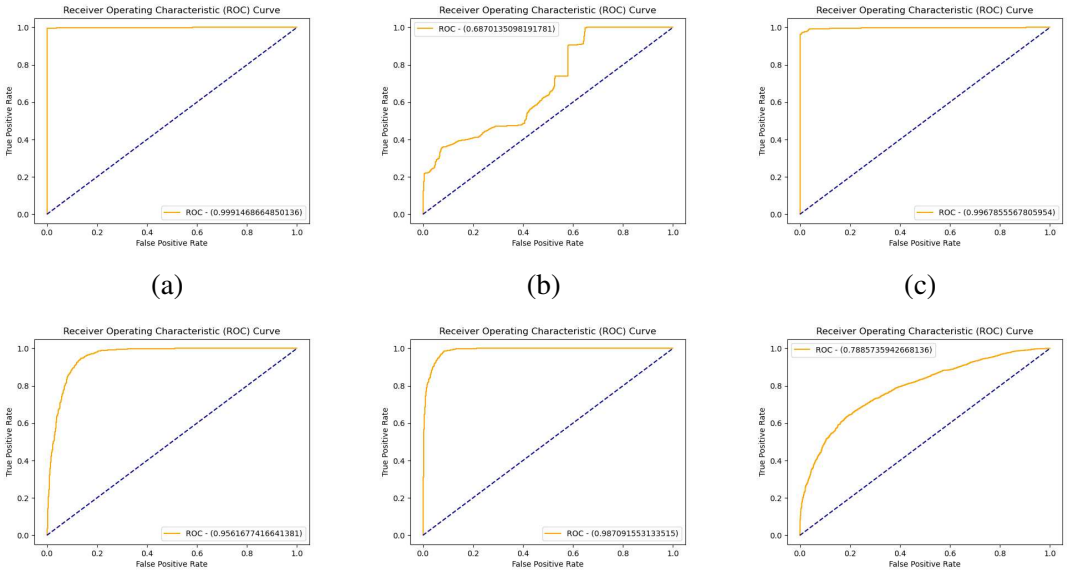


Figure 4.3: Comparison of AUROC curves for Entropy metric using Flowers102 Dataset. Each column represent AUROC curves for ID vs Near OOD, ID vs Far OOD, Near vs Far OOD. First row contains plots for ViT classifier baseline results followed by the second row which shows our work.

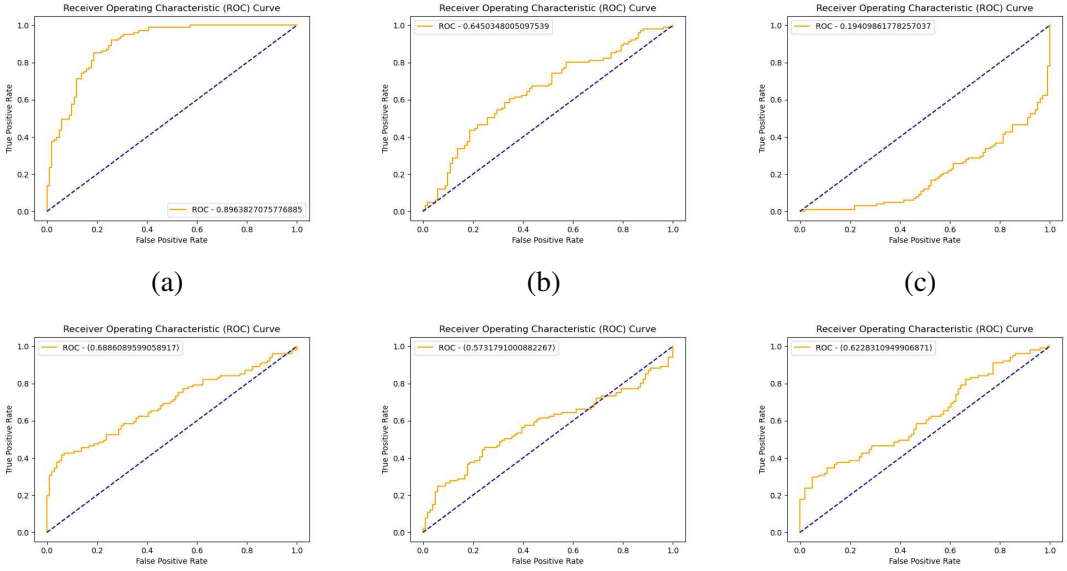


Figure 4.4: Comparison of AUROC curves for Likelihood metric using Flowers102 Dataset. Column wise each column (a-c d-f)refers to the AUROC curves for ID vs Near OOD, ID vs Far OOD, Near OOD vs Far OOD. First row shows DCGANVAE baseline results while the second row shows our proposed decoder results

value of entropy, the far OOD sample leads to a high value of entropy. Interestingly, the entropy of near-OOD sample lies in between ID and far-OOD sample for all the datasets. We observe correlating trends in MSP as well.

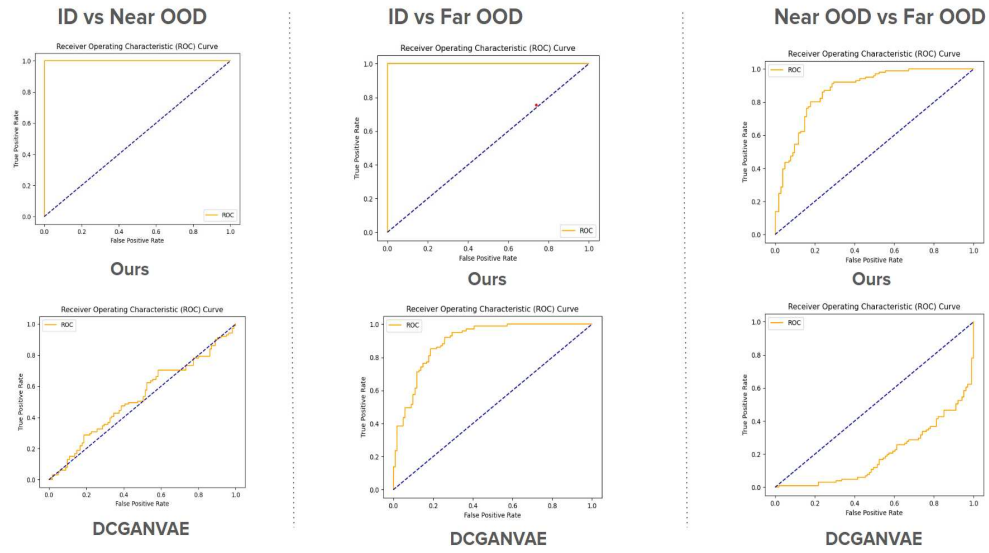
Table 4.3: Comparison of density based scores using AUROC, AUPR, FPR@95 for Near OOD Detection using DCGANVAE as the baseline

Dataset	**Model**	**AUROC**	**AUPR**	**FPR@95**
Flowers102	Baseline	0.527	0.516	0.98
Flowers102	Ours	**1**	**1**	**0.01**
CUB200	Baseline	0.4411	0.6175	0.8613
CUB200	Ours	**1**	**1**	**0.01**

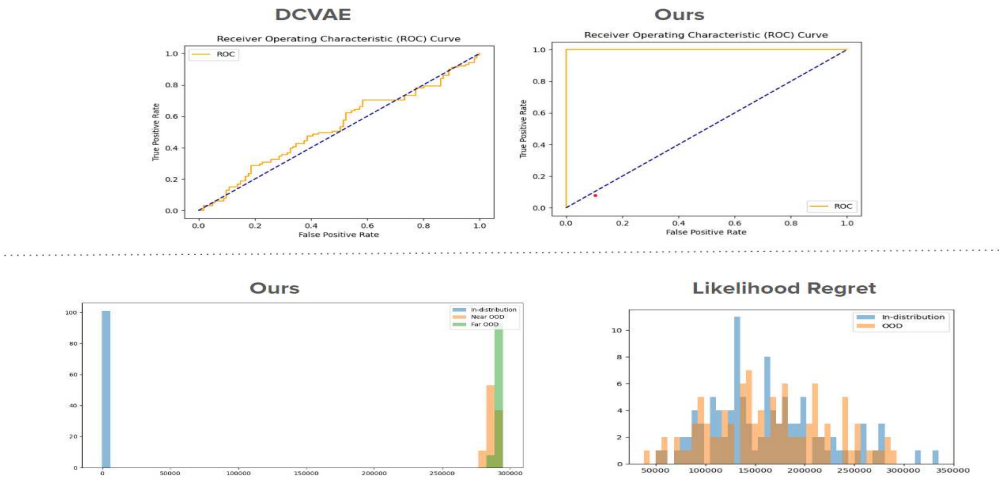
Table 4.4: Comparison of density based scores using AUROC, AUPR, FPR@95 for Far OOD Detection using DCGANVAE as the baseline

Dataset	**Model**	**AUROC**	**AUPR**	**FPR@95**
Flowers102	Baseline	0.645	0.617	0.861
Flowers102	Ours	**1**	**1**	**0.01**
CUB200	Baseline	0.8815	0.7325	0.4752
CUB200	Ours	**1**	**1**	**0.01**

We observe that UMAP preserves the global structure of the data as well as faster than TSNE. Also UMAP provides interpretable results unlike TSNE for out of distribution samples. The ID, Near OoD as well as far OoD samples seems to be vastly separated in the UMAP diagrams 4.6. we conclude that class token which contains the class level information along with the decoder architecture captures the global patch information and therefore fine grained changes are well captured. This is well reflected in the entropy and MSP plots.

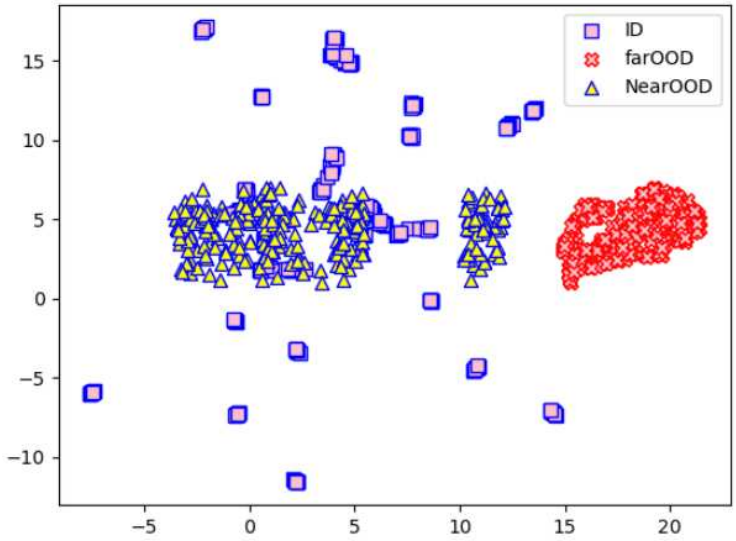


(a) Comparison of AUCROC on Flowers102 using SRGAN Decoder

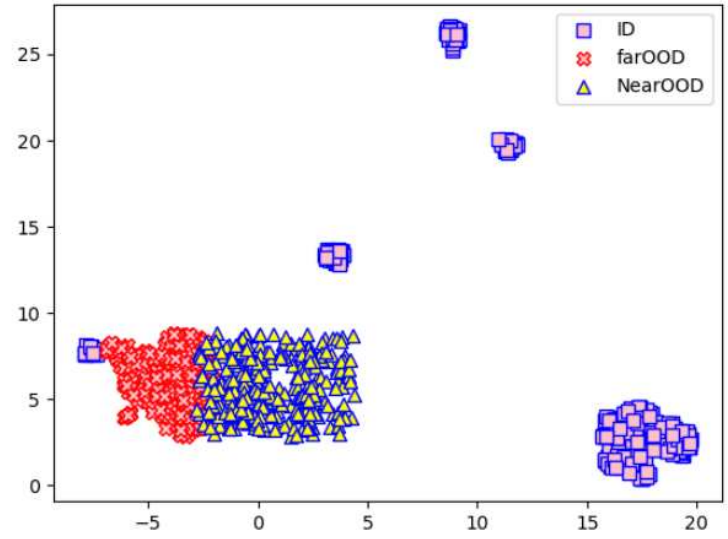


(b) Comparison of Likelihood Regret method on Flowers102 dataset using SRGAN Decoder

Figure 4.5: Comparison of SRGAN Decoder on likelihood based performance metrics

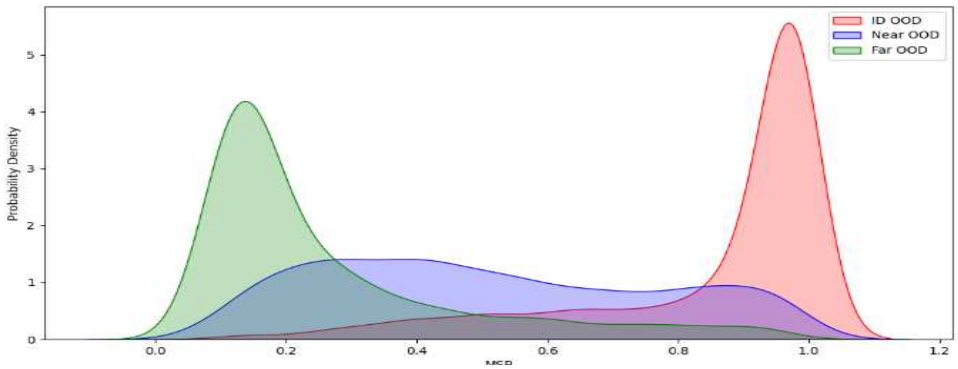


(a) Flowers102 Data

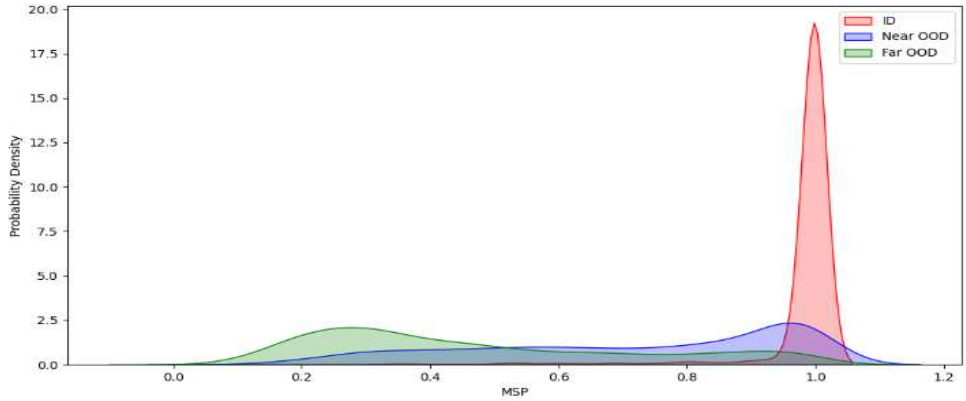


(b) CUB200 Data

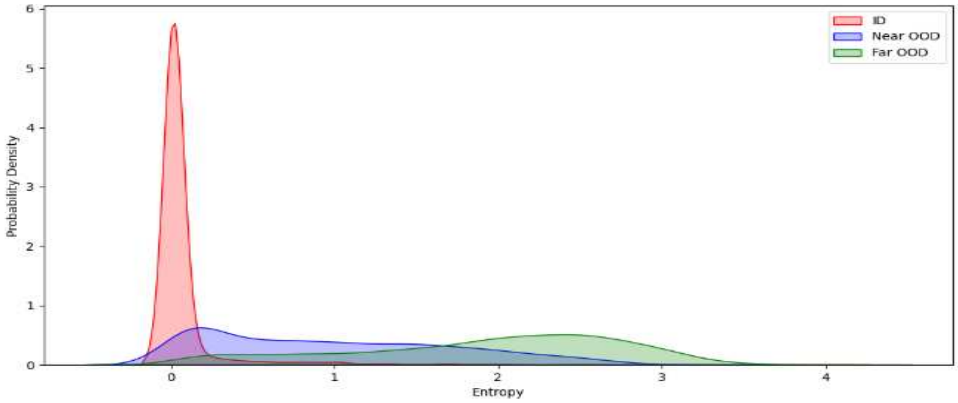
Figure 4.6: UMAP of ID, Near OoD and Far OoD embeddings for Flowers and CUB200 Dataset



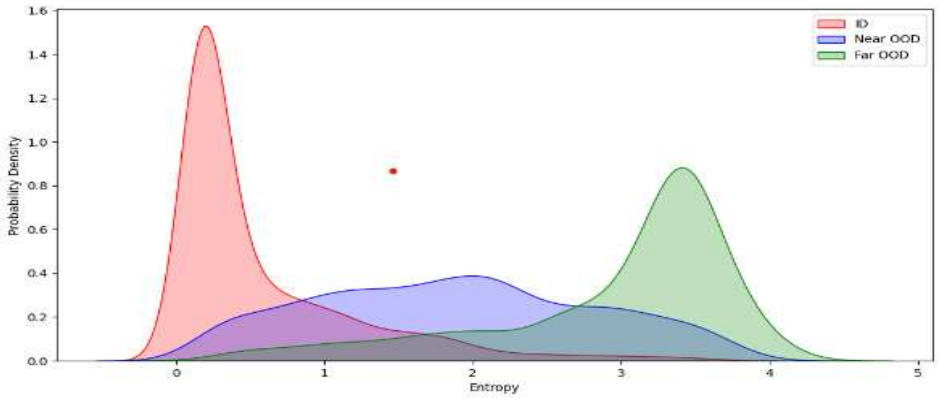
(a) MSP of CUB200 Dataset



(b) MSP of Oxfordflowers102 Dataset



(c) Entropy of CUB200 Dataset



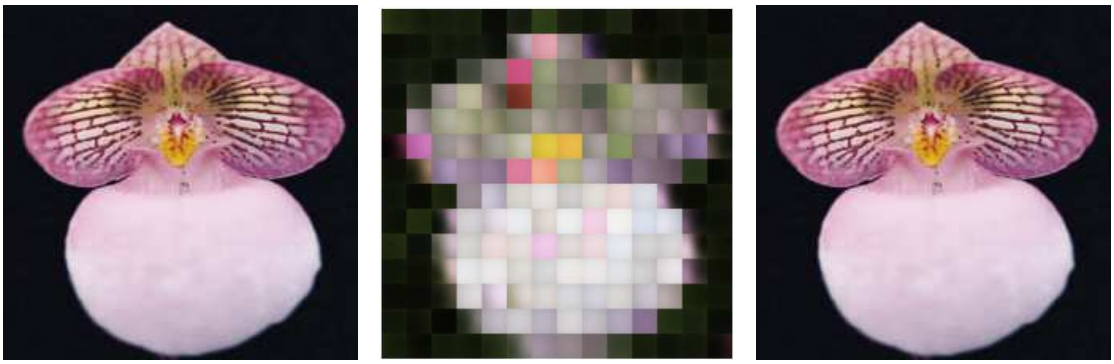
(d) Entropy of Oxfordflowers102 Dataset

Figure 4.7: Comparison of Entropy and MSP for OOD Detection in Flowers and CUB dataset

CHAPTER 5

Conclusion and Future Work

In this work we propose an architecture which can detect out of distribution samples in fine grained datasets. We show that our work outperforms the existing work on near OoD Detection on datasets like Oxford Flowers 102, and CUB 200. Our work covers two different types of OoD detection like classification and density based techniques. Our proposed pipeline which contains a ViT based VAE architecture along with a classifier does patch level disentanglement which captures the information contained in each patch. We also show that using ESRGAN Generator as the decoder network performs better on density based detection methods but performs poorly at classification based methods. We are optimistic that our work will help detect out of distributions samples in real world cases like medical imaging and cancer detection.



(a) Flowers original data (b) Output from custom decoder (c) Output of SRGAN Decoder

Figure 5.1: Comparison of our proposed pipeline with custom decoder vs SRGAN Decoder

Although our work does better than other proposed baselines but it can be extended further based on the results obtained from SRGAN Decoder. We observe that perceptual loss along with MSE loss gives similar reconstructions than the original image indicating that the learned disentangled representations have successfully captured the style and artistic qualities however the global features captured by the model is not good enough for an accurate classification. In future, this work can be extended in multimodality and several other domain specific settings.

REFERENCES

1. (). Technical report.

2. **Abnar, S.** and **W. Zuidema** (2020). Quantifying attention flow in transformers. URL https://arxiv.org/abs/2005.00928.

3. **Chen, X.**, **Y. Duan**, **R. Houthooft**, **J. Schulman**, **I. Sutskever**, and **P. Abbeel** (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. URL https://arxiv.org/abs/1606.03657.

4. **Cultrera, L.**, **L. Seidenari**, and **A. Del Bimbo**, Leveraging visual attention for out-of-distribution detection. *In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. 2023.

5. **Denouden, T.**, **R. Salay**, **K. Czarnecki**, **V. Abdelzad**, **B. Phan**, and **S. Vernekar** (2018). Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance. URL https://arxiv.org/abs/1812.02765.

6. **Dosovitskiy, A.**, **L. Beyer**, **A. Kolesnikov**, **D. Weissenborn**, **X. Zhai**, **T. Unterthiner**, **M. Dehghani**, **M. Minderer**, **G. Heigold**, **S. Gelly**, *et al.* (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

7. **Fort, S.**, **J. Ren**, and **B. Lakshminarayanan** (2021). Exploring the limits of out-of-distribution detection. URL https://arxiv.org/abs/2106.03004.

8. **Goodfellow, I. J.**, **J. Pouget-Abadie**, **M. Mirza**, **B. Xu**, **D. Warde-Farley**, **S. Ozair**, **A. Courville**, and **Y. Bengio** (2014). Generative adversarial networks. URL https://arxiv.org/abs/1406.2661.

9. **Hendrycks, D.** and **K. Gimpel** (2018). A baseline for detecting misclassified and out-of-distribution examples in neural networks. URL https://arxiv.org/abs/1610.02136.

10. **Higgins, I.**, **L. Matthey**, **A. Pal**, **C. Burgess**, **X. Glorot**, **M. Botvinick**, **S. Mohamed**, and **A. Lerchner**, beta-VAE: Learning basic visual concepts with a constrained variational framework. *In International Conference on Learning Representations*. 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

11. **Jiang, W.**, **Y. Ge**, **H. Cheng**, **M. Chen**, **S. Feng**, and **C. Wang** (2023). Read: Aggregating reconstruction error into out-of-distribution detection. URL https://arxiv.org/abs/2206.07459.

12. **Johnson, J.**, **A. Alahi**, and **L. Fei-Fei** (2016). Perceptual losses for real-time style transfer and super-resolution. *CoRR*, **abs/1603.08155**. URL http://arxiv.org/abs/1603.08155.

13. **Kim, H.** and **A. Mnih** (2019). Disentangling by factorising. URL https://arxiv.org/abs/1802.05983.

14. **Koner, R.**, **P. Sinhamahapatra**, **K. Roscher**, **S. Günnemann**, and **V. Tresp** (2021). Oodformer: Out-of-distribution detection transformer. *arXiv preprint arXiv:2107.08976*.

15. **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.

16. **Liu, L.**, **J. Li**, **L. Niu**, **R. Xu**, and **L. Zhang** (2021). Activity image-to-video retrieval by disentangling appearance and motion. *Proceedings of the AAAI Conference on Artificial Intelligence*, **35**, 2145–2153.

17. **Macêdo, D.**, **T. I. Ren**, **C. Zanchettin**, **A. L. Oliveira**, and **T. Ludermir**, Entropic out-of-distribution detection. *In 2021 international joint conference on neural networks (IJCNN)*. IEEE, 2021.

18. **Mukhoti, J.**, **T.-Y. Lin**, **B.-C. Chen**, **A. Shah**, **P. H. Torr**, **P. K. Dokania**, and **S.-N. Lim**, Raising the bar on the evaluation of out-of-distribution detection. *In Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023.

19. **Ren, J.**, **P. J. Liu**, **E. A. Fertig**, **J. R. Snoek**, **R. Poplin**, **M. DePristo**, **J. Dillon**, and **B. Lakshminarayanan**, Likelihood ratios for out-of-distribution detection. 2019. URL `https://arxiv.org/abs/1906.02845`.

20. **Sim, M.**, **J. Lee**, and **H.-J. Choi**, Attention masking for improved near out-of-distribution image detection. *In 2023 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2023.

21. **Singla, S.**, **N. Murali**, **F. Arabshahi**, **S. Triantafyllou**, and **K. Batmanghelich**, Augmentation by counterfactual explanation-fixing an overconfident classifier. *In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2023.

22. **Sun, Y.**, **C. Guo**, and **Y. Li**, React: Out-of-distribution detection with rectified activations. *In* **M. Ranzato**, **A. Beygelzimer**, **Y. Dauphin**, **P. Liang**, and **J. W. Vaughan** (eds.), *Advances in Neural Information Processing Systems*, volume 34. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/01894d6f048493d2cacde3c579c315a3-Paper.pdf`.

23. **Sun, Y.**, **Y. Ming**, **X. Zhu**, and **Y. Li**, Out-of-distribution detection with deep nearest neighbors. *In International Conference on Machine Learning*. PMLR, 2022.

24. **Tafvizi, A.**, **B. Avci**, and **M. Sundararajan** (2022). Attributing auc-roc to analyze binary classifier performance. URL `https://arxiv.org/abs/2205.11781`.

25. **Vaswani, A.**, **N. Shazeer**, **N. Parmar**, **J. Uszkoreit**, **L. Jones**, **A. N. Gomez**, **L. Kaiser**, and **I. Polosukhin** (2023). Attention is all you need. URL `https://arxiv.org/abs/1706.03762`.

26. **Wang, X.**, **H. Chen**, **S. Tang**, **Z. Wu**, and **W. Zhu** (2024). Disentangled representation learning. URL `https://arxiv.org/abs/2211.11695`.

27. **Wei, H.**, **R. Xie**, **H. Cheng**, **L. Feng**, **B. An**, and **Y. Li** (2022). Mitigating neural network overconfidence with logit normalization. URL `https://arxiv.org/abs/2205.09310`.

28. **Xiao, Z.**, **Q. Yan**, and **Y. Amit** (2020). Likelihood regret: An out-of-distribution detection score for variational auto-encoder. *Advances in neural information processing systems*, **33**, 20685–20696.

29. **Yang, J.**, **K. Zhou**, **Y. Li**, and **Z. Liu** (2024). Generalized out-of-distribution detection: A survey. URL https://arxiv.org/abs/2110.11334.

30. **Yang, Y.**, **R. Gao**, and **Q. Xu** (2022). Out-of-distribution detection with semantic mismatch under masking. URL https://arxiv.org/abs/2208.00446.

31. **Zhang, C.**, **M. Zhang**, **S. Zhang**, **D. Jin**, **Q. Zhou**, **Z. Cai**, **H. Zhao**, **X. Liu**, and **Z. Liu** (2022). Delving deep into the generalization of vision transformers under distribution shifts. URL https://arxiv.org/abs/2106.07617.

32. **Zhang, J.**, **L. Gao**, **B. Hao**, **H. Huang**, **J. Song**, and **H. Shen** (2023). From global to local: Multi-scale out-of-distribution detection. URL https://arxiv.org/abs/2308.10239.

33. **Zhang, Y.**, **W. Chen**, **Z. Zhu**, **D. Qin**, **L. Sun**, **X. Wang**, **Q. Wen**, **Z. Zhang**, **L. Wang**, and **R. Jin** (2024). Addressing concept shift in online time series forecasting: Detect-then-adapt. URL https://arxiv.org/abs/2403.14949.