Action Recognition in Egocentric Videos

by

Sagar Verma

Under the Supervision of

Dr. Chetan Arora, IIIT Delhi

Indraprastha Institute of Information Technology, Delhi

July, 2017

Keywords: Deep Learning, Computer Vision, Egocentric Videos, CNN, LSTM, GoPro, Google Glass, Microsoft Hololens, Action Recognition

Action Recognition in Egocentric Videos

Sagar Verma
IIIT-D-MTech-CS-GEN-15-056

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science

to

Indraprastha Institute of Information Technology Delhi

July, 2017

Certificate

This is to certify that the thesis titled **"Action Recognition in Egocentric Videos"** submitted by **Sagar Verma** to the Indraprastha Institute of Information Technology Delhi, for the award of the *Master of Technology* in *Computer Science & Engineering*, is an original research work carried out by him under my supervision. In my opinion, the thesis has reached the standards fulfilling the requirements of the regulations relating to the degree. The results contained in this thesis have not been submitted in part or full to any other University or Institute for the award of any degree/diploma.

Dr. Chetan Arora

Dept. of Computer Science

IIIT Delhi

Abstract

With an increase in usage and availability of wearable devices like GoPro, Microsoft Hololens, Google Glass, etc, egocentric video analysis has become essential. An interesting application is action recognition in egocentric videos. Research has been performed on action recognition in first person(egocentric) videos.

First person action recognition is a hard problem given that first person videos are shaky, have limited hand-object interaction, and have limited publicly available datasets. Most of the existing research uses hand-crafted features to learn actions which works best for a given domain. First person videos have two types of actions. First, where hand-object interactions are present and the other one, where no such interactions are present. Current methods can only be used to recognise any one type of action but not both using a single method.

This research proposes a novel action recognition method to recognise two types of actions, one where hand-object interaction is present and other where no such interactions are present. Further, a new dataset named IIITD Plumbing dataset is introduced which provides large number of videos, objects, and actions. The proposed system makes use of spatio-temporal information captured from raw frames. We also introduce a new method to perform activity recognition that learns grammar from learned actions.

Acknowledgements

I sincerely thank my supervisor Dr. Chetan Arora, IIIT Delhi for giving me the opportunity to work on this project and also for his constant supervision and valuable guidance during the course of the thesis. I would like to thank all the colleagues at Computer Vision and Machine Learning lab with whom I have shared a major portion of my time here at IIIT Delhi. Their friendly advice and professional wisdom has always helped me gain new perspective of my research work. This section can not be complete without a vote of thanks to academic department for their help and never ending support.

Contents

List of Figures

List of Tables

Chapter 1

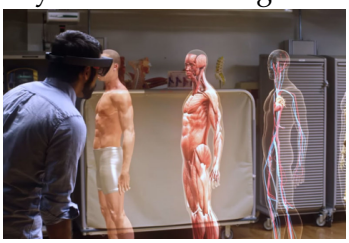
Introduction

1.1 Overview and Research Motivation

(a) Skydive Logging using GoPro [2]

(b) Mobility Assistance using Pivothead [6]

(c) Policeman using Google Glass [1]



(d) Hololens for Agumented Reality [3]

Figure 1.1: Commercially available egocentric cameras and their usages.

With improvements in technology and usability, wearable devices like GoPro [2], [6], Google Glass [1], Microsoft sensecam [4], Microsoft Hololens [3], etc., are becoming ubiquitous. The cameras are typically harnessed to a wearer's head giving a first person perspective. We refer to such cameras as egocentric cameras. The unique perspective of egocentric camera, as well as, commonly available always-on feature, makes use of such cameras compelling in applications like extreme sports, law enforcement, life logging, home automation, virtual reality, augmented reality and assistive vision.

In egocentric context, extensive research has been done on the applications like object recognition [15, 44, 43], activity recognition [16, 15, 34, 37, 39, 46, 54, 55, 67, 33], video summarization [8, 28, 32, 63] etc. Along with these conventional applications, some very interesting and unique applications have also been proposed. Some notable examples include understanding social interactions [14, 65], privacy control [40], biometrics [17, 61], gaze detection [30], wearer localization [52, 45], and force estimation [53]. Given the applications of egocentric perspective in augmented/virtual reality, researchers have also been working on hand pose and grip recognition problems [29, 10, 44, 43].

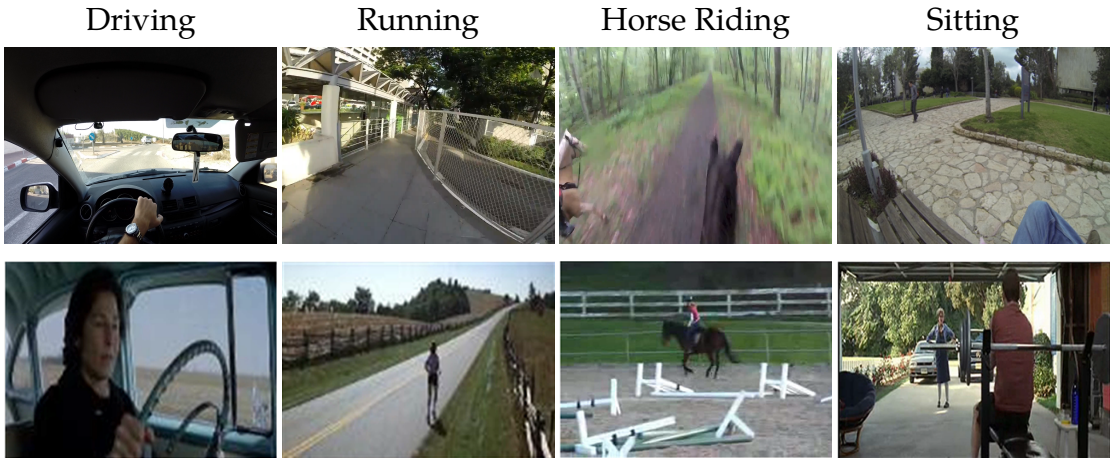


Figure 1.2: Different actions in first person and third person perspective. Top row shows some first person actions and bottom row shows same action in third person perspective.

The first person perspective of an egocentric camera often breaks common assumptions inherent in conventional computer vision techniques. For example, in the context of action recognition, while conventional third person action recognition techniques use the pose of the actor as an important cue, for first person action recognition, the egocentric camera does not even see the actor. The algorithms for first person action recognition, therefore, must rely on secondary cues such as hands, objects and motion profile. Figure 1.2 shows the two perspective, top row shows first person perspective taken from HUJI [42] data set, and bottom row shows third person perspective taken from Hollywood2 [27] dataset.

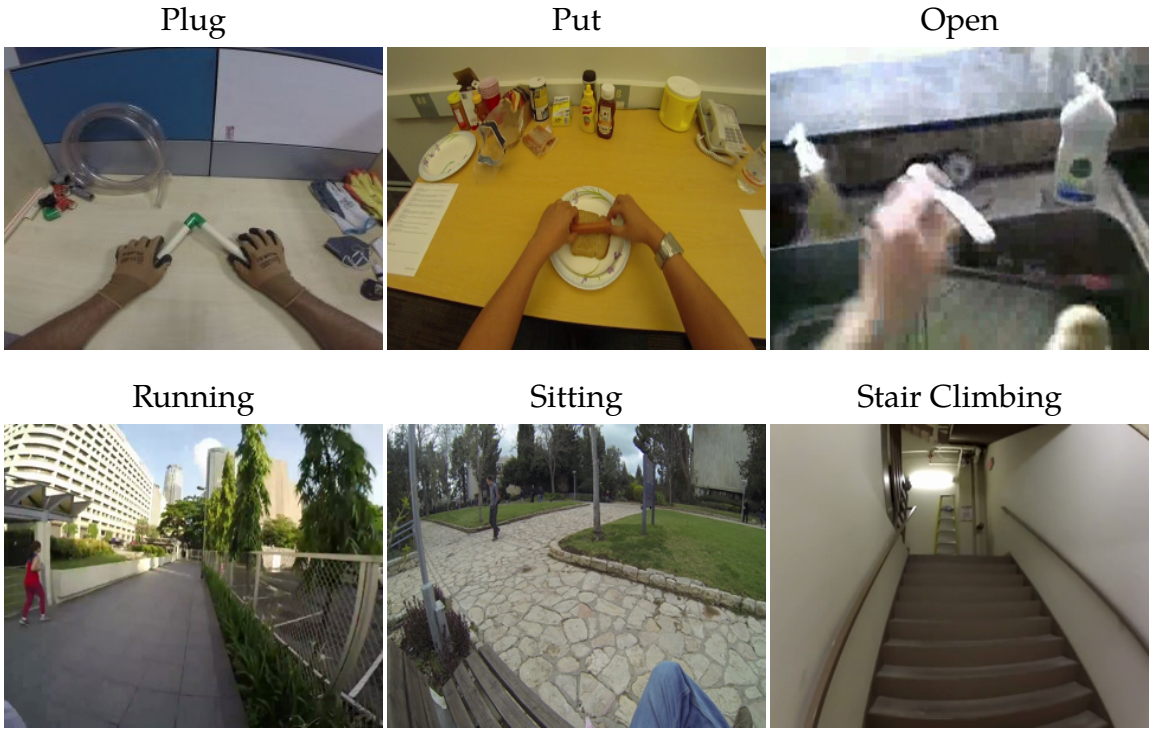


Figure 1.3: Sample actions from two action categories; top row show actions where some form of hand-object interaction is present, and bottom row shows actions without any hand-object interaction.

First person actions can be categorized into two categories, one where some form of hand object interaction is present and other where such interactions are not present. Certain approaches that use object and hand recognition to recognize actions works well in the first category and not in the second category. From Figure 1.3 it can be seen that actions where hand-object interaction is present detecting such interactions may provide important cues for action recognition. Whereas actions that don't have such interactions, no such cues can be captured. The objective of this research is to use one method to recognize both types of actions thus extending the reachability of a single method over more and different types of actions.

Most of the egocentric videos capture some form of activity. These activities can again be seen as a sequence of actions. One other way is to think of an activity as an action which can be composed into many sub-actions and the sequence of these sub-actions uniquely identifies the top level action. When posed like this problem becomes hierarchical video classification. These type of activities may occur where actions being performed have some sort of hand-object interactions.

Most of the above-mentioned problems are being solved for both first person and third person perspective. Pipeline for action recognition for both the perspective is similar with a slight difference in how each stage of the pipeline is implemented. A typical egocentric action recognition pipeline consists of following steps:

- **Data collection:** Data is collected using either head or chest mounted camera or a pivothead camera.
- **Preprocessing:** Egocentric videos are shaky and require stabilization algorithms to remove head movements.
- **Feature extraction:** For learning purpose some form of feature extraction is required.
- **Learning:** A model is learned on extracted features.
- **Prediction:** Learned model is used to do action recognition.

1.2 Literature Review

1.2.1 Third Person Action Recognition

In third person action recognition, a lot of work has been done in the area of feature extraction and learning. These works can be classified into two categories, one is where hand crafted features are used to learn some sort of model and other is automatically learning such features using some form of learning method. We discuss some of the relevant work in each of the categories,

Hand Crafted Features for Third Person Action Recognition

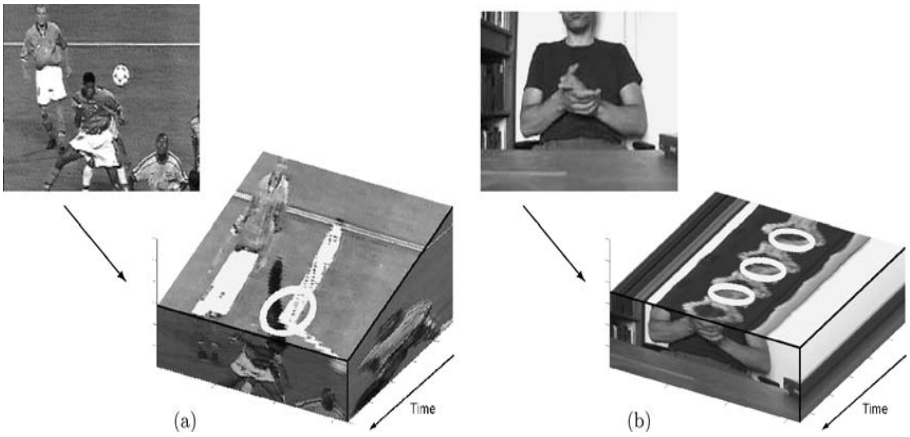


Figure 1.4: Result of detecting the strongest spatio-temporal interest points from [27].

For the conventional third person action recognition task, the standard approach is to learn and match visual features from video frames based on key-points and descriptors. These features are generally hand-tuned. Some notable contributions in this area include STIP [27], 3D-SIFT [47], HOG3D [23], extended SURF [62], and

Local Trinary Patterns [64]. Recently the methods using appearance and motion information around densely sampled point trajectories have also shown promising results [59, 60, 18, 24].

Space Time Interest Points(STIP) [27] method uses local image features or interest points which provides compact and abstract representations of patterns in an image. STIP features often reflect interesting events that can be used for a compact representation of video data as well as for interpretation of spatiotemporal events. To detect spatiotemporal events, the Harris and Forstner interest point operators are used which detects local structures in space-time where the image values have significant local variations in both space and time. The method estimates the spatiotemporal extents of the detected events by maximizing a normalized spatiotemporal Laplacian operator over spatial and temporal scales. The result of detecting the strongest spatiotemporal interest points in a football sequence with a player heading the ball (a) and in a hand clapping sequence (b). From the temporal slices of space-time volumes are shown in Figure 1.4, it is evident that the detected events correspond to neighborhoods with high spatiotemporal variation in the image data or 'space-time corners'. The method proposed in [47] transfers 2D descriptors into 3D descriptors method by introducing gradient calculation in the third direction, i.e., time.

Figure 1.5: Sample frames from video sequences classified using [23].

In [23] a novel local descriptor for video sequences is proposed. This descriptor is based on histograms of oriented 3D spatiotemporal gradients. Their method computes 3D gradients for arbitrary scales, and is a memory-efficient algorithm based on integral videos and performs well in many action recognition datasets. Action recognition method proposed in [64] is based on combining the effective description properties of Local Binary Patterns with the appearance invariance and adaptability of patch matching based methods. The method is extremely efficient and works for the real-time use of simultaneous recovery of human action of several lengths and starting points.

In [59] feature trajectories are shown to be efficient for representing videos. They are extracted using the KLT tracker or matching SIFT descriptors between frames. They sample dense points from each frame and track them based on displacement information from a dense optical flow field. They use Bag-of-Words

approach to represent these descriptors for action classification task. They extend their work in [60] by removing camera motion from trajectories. The method matches feature points between frames using SURF descriptors and dense optical flow. These matches are, then, used to robustly estimate a homography with RANSAC. Human motion is in general different from camera motion and generates inconsistent matches. To improve the estimation, a human detector is employed to remove these matches. Given the estimated camera motion, trajectories consistent with it are removed. This estimation is used to cancel out camera motion from the optical flow. This significantly improves motion-based descriptors, such as HOF and MBH.

In [18] authors have established that adequately decomposing visual motion into dominant and residual motions, both in the extraction of the space-time trajectories and for the computation of descriptors, significantly improves action recognition algorithms. They have designed a new motion descriptor, the DCS descriptor, based on differential motion scalar quantities, divergence, curl and sheer features. It captures additional information on the local motion patterns which gives better action recognition results.

Deep Learned Features for Third Person Action Recognition

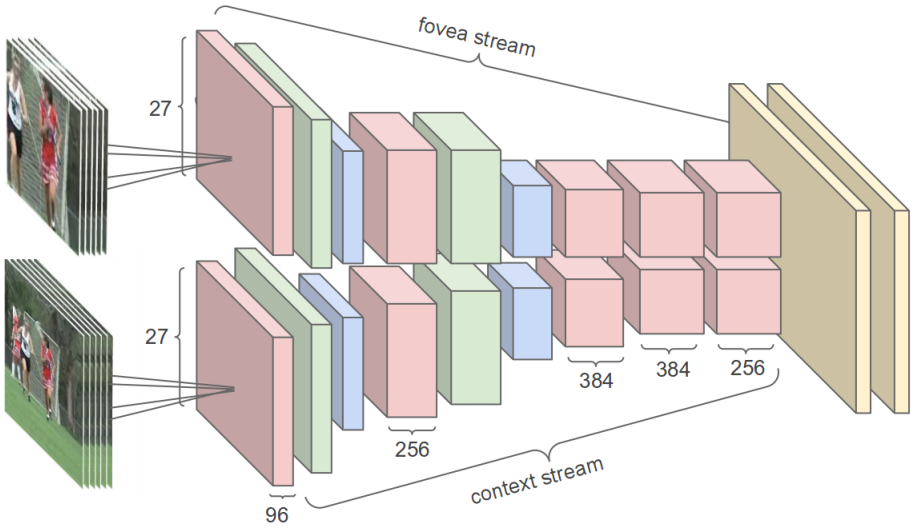


Figure 1.6: Multiresolution CNN architecture proposed in [21].

Deep learned features for action recognition have also been explored. In the third person action recognition, context is often useful and therefore, using only image based features can give reasonable performance [21]. Therefore, many works in this area like Convolutional RBMs [56], 3D ConvNets [19], C3D ConvNet [57], take only video frame as input. Recently, two-stream ConvNets [48] introduced spatial as well as flow streams for action recognition. Arguably, change in the actor's pose, background information and appearance of objects over a sequence of frames in a video are strong indicators about the action being performed. The recent works in the area have also tried to learn these patterns over a video [21, 12]. The work closest to this research is [35], where the authors have similarly used a CNN-LSTM model for third person actions. They have trained and tested their model on a huge dataset of 1 million sports video from 1000 action categories downloaded from

YouTube. The action categories are widely different in appearance which results in the authors achieving a similar performance using even the single frame as well as without the use of LSTMs.

Convolutional Neural Networks (CNNs) is used for image recognition problems in [21]. They have done an extensive empirical evaluation of CNNs on largescale video classification on a dataset of 1 million YouTube videos belonging to 487 classes. They are first to propose multiple approaches for extending the connectivity of a CNN in the time domain to take advantage of local spatiotemporal information and suggest a multi resolution, foveated architecture as a promising way of speeding up the training. Their methods have passed the traditional state of the art by huge margins.

The problem of learning good features for understanding video data is solved in [56]. They introduce a model that learns latent representations of image sequences from pairs of successive images. The convolutional architecture of their model allows it to scale to realistic image sizes whilst using a compact parametrization. They also use their model to extract low-level motion features in a multi-stage architecture for action recognition on the KTH and Hollywood2 datasets.

The method presented in [19] uses 3D convolution for the fully automated recognition of actions in the uncontrolled environment. 3D convolutional model extracts features from both spatial and temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. The developed model generates multiple channels of information from the input frames, and the final feature representation is obtained by combining information from all channels. The developed model is used to recognize human actions in the real-world environment.

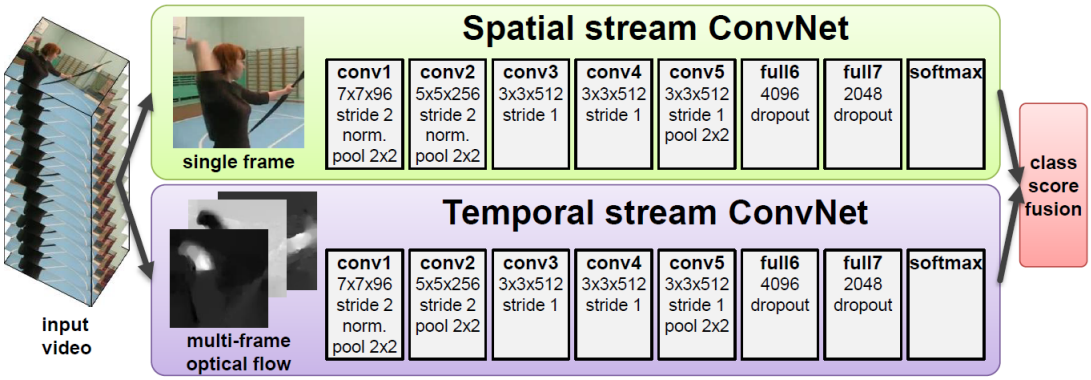


Figure 1.7: Two stream network for action recognition presented in [48].

In order to capture the complementary information on appearance from still frames and motion between frames [48] presents a two stream network. Spatial stream capture spatial information from raw RGB frames and temporal stream capture temporal information from the dense optical flow.

To learn temporal features [12, 35] use LSTM which takes the latent feature from CNN network. [12] use LSTM and CNN network for action recognition, image captioning and video captioning. They use Sports-1M dataset for action recognition which is a very large dataset and makes it possible to train multi-layered LSTM networks.

1.2.2 First Person Action Recognition

Like third person action recognition, first person action recognition can also be divided into two categories based on how features are computed.

Hand Crafted Features for First Person Action Recognition

Figure 1.8: Background and foreground segmentation in [15].

Most of the earlier works on first person action recognition use hands and objects as important cues in their pipeline [15, 44, 43, 16, 34, 37, 39, 46, 54, 55]. However, recently, some egocentric researchers have been able to show, in a limited context, that when the action does not involve any handled object, features based on optical flow alone can be used for the first person action recognition task [41, 22].

In [15] a probabilistic generative model is used for simultaneously recognizing daily actions and predicting gaze locations in egocentric videos. They focus on activities requiring eye-hand coordination and model the spatiotemporal relationship between the gaze point, the scene objects, and the action label. Their model captures the fact that the distribution of both visual features and object occurrences in the vicinity of the gaze point is correlated with the verb-object pair describing the action. It explicitly incorporates known properties of gaze behavior from the psychology literature, such as the time delay between fixation and manipulation events. They present an inference method that can predict the best sequence of gaze locations and the associated action label from an input sequence of images. SIFT based recognition method is used in [44] to do object recognition of handheld objects. [43] use a bottom-up motion-based approach to segment out foreground

objects. They compute dense optical flow and fit it into multiple affice layers and then use a max-margin classifier to combine motion with empirical knowledge of object location and background movement as well as temporal cues of support region and color appearance.

Learning object models from the egocentric video is introduced in [16]. For each action sequence, only the names of the objects which are present within it is known and no other knowledge regarding the appearance or location of objects is known. An unsupervised bottom up segmentation method is used to capture the structure of the first person domain to partition each frame into hand, object, and background classes. This work enables researchers to work on datasets which don't have frame level and in frame location level labeling of objects. In [34] an attention based method is used to get object categories, this ensures that objects that are occluded by hands also get detected.

First person eye movement and ego-motion are also important features to learn about actions being performed. This is shown in [37] where they use an "inside-out" camera to capture eye movement. Frames from both cameras "inside-out" and egocentric are used to capture ego-motion and perform activity recognition. In [39] ADL data set is introduced. They use a temporal pyramid which is an extension of the spatial pyramid to do object recognition. The problem of recognizing interaction-level human activities from a first-person viewpoint is presented in [46]. Temporal segmentation of human motion into actions for activity recognition is done in [54]. RGB frames along with IMUs data is used to do both supervised and unsupervised temporal segmentation. Using a 3-level Dynamic Bayesian Network and temporal templates activity recognition is done in low-resolution images. [55] have presented this method which is computationally efficient and can be used in low-end devices. [22] use a motion-based histogram and unsupervised

learning algorithms to cluster video content.

Deep Learned Features for First Person Action Recognition

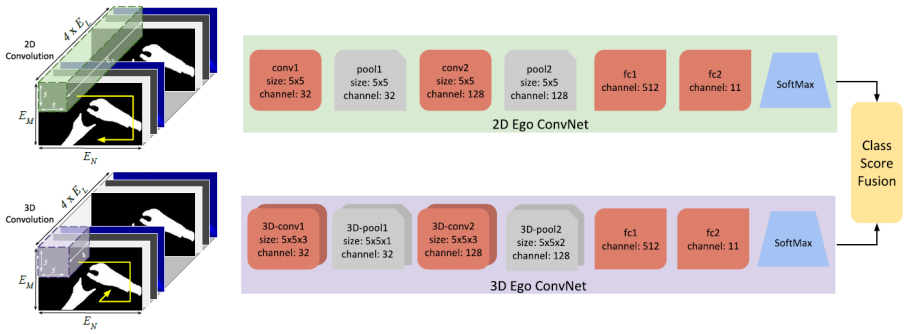


Figure 1.9: Two stream, 2D and 3D convolutional network with egocentric cues as input given in [51].

Recent work that involves learning of features using convolutional neural networks are [51, 33]. Singh *et al.* [51] have used hand pose, head motion and salient motion as input to a two layered convolutional network. Ma *et al.* [33] have used a multi task learning framework using hand appearance and object attributes for activity recognition into 71 classes. The action labels correspond to a combination of action and object attributes (e.g. 'take-cheese' and 'take-bread' are two different classes in their framework).

The method proposed by [51] uses convolutional neural networks (CNNs) for an end to end learning and classification of wearer's actions. The proposed network makes use of egocentric cues by capturing hand pose, head motion and saliency map. They use the two stream network of [48] along with a third stream that does 3D convolution on egocentric cues. The third stream is shown in Figure 1.9. The output of all streams are combined using an SVM. [33] have used a twin

stream network architecture, where one stream analyzes motion information and other analyses appearance information. Appearance stream encodes prior knowledge of the egocentric domain by explicitly training the network to segment hands and localize objects.

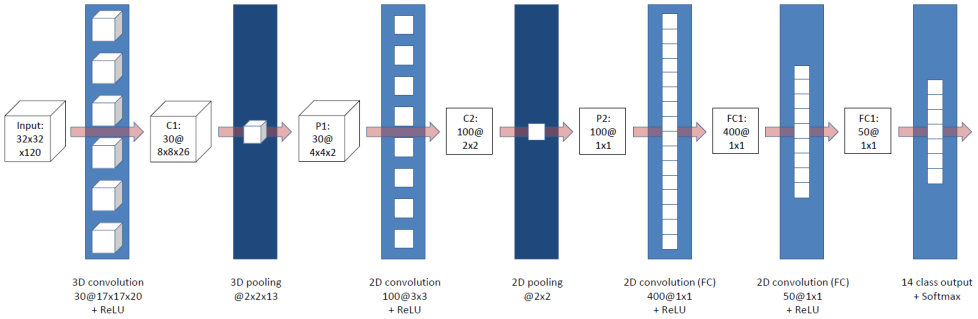


Figure 1.10: 3D convolutional network which takes sparse flow as input.

For the first person actions which do not involve handled objects, researchers have typically used patterns of wearer's head motion for the recognition task. Kitani *et al.* [22] have suggested motion based histograms generated from the optical flow to learn the first person actions. Singh *et al.* [50] have proposed trajectory aligned features along with simple egocentric cues for first person action recognition. [42] has used compact CNN architecture for recognizing long term actions without any handled object. [50] use a representation of the first person actions derived from feature trajectories. The features are simple to compute using standard point tracking and does not assume segmentation of hand/objects or recognizing object or hand pose. They train a bag of words classifier with the proposed features. Classifying video using just sparse flow is done in [42]. They compute sparse flow over 32×32 blocks and use 3D CNN to classify into actions.

Recognising both Categories of Actions

Reliance on the state of the art on specific egocentric cues makes them restricted to particular action categories focussed upon in the work. For example, it is not clear how the sparse optical flow based features suggested in [41, 42] can be used for actions like stir and spread. on the other hand, it is not obvious to extend the techniques of [33, 51], which detects hands and objects, explicitly or implicitly, to the actions like walk and run. Our focus in this research is on developing a generic technique, which can handle various kinds of first person action categories proposed so far, with a single universal model.

1.2.3 Avtivity Recognition

Most of the works in activity recognition from action labels use clustering or HMM to learn a model. [26] have done fine grained classification of actions into spatial and temporal elements using unsupervised learning. They have used discriminative clustering algorithm which separates action-related segments from background actions. [31] uses sensor data to identify temporal patterns and then do actions to activity hierarchical classification. [38] have done structured activity recognition using a language based approach, the proposed algorithm is called Helix. It first generates an initial vocabulary using unlabeled sensor readings, followed by iteratively combining statistically correlated sub-activities across sensor dimensions and grouping similar activities together to discover higher level activities. [58] have presented a two-layer hierarchical model in which activities consist of a sequence of actions which are clustered into a group of actions. [36] have used an HMM based approach that uses threshold and voting to automatically and effectively segment and recognize complex activities.

1.3 Research Contributions

Our focus is on the recognition of wearer's action from an egocentric video. Unlike most of the existing work, our objective is to develop a technique which can target a variety of action categories involving object handling as well as without. First person actions such as take, pour, spread, stir etc. typically involve handling some object. While, the other first person actions such as walk, run etc. may not have any such handled objects. Further classification can be done in terms of length of the action. For example, actions, like walking or running, may span several minutes, while many of the actions involving object handling such as take or pour last only a few seconds. We would like the proposed technique to be agnostic to the length of the action and detect both categories of actions.

First person action recognition from egocentric videos is an independent and different problem than the conventional action recognition. As described earlier, while the actor's pose is the most important cue in conventional third person action recognition, the same is simply not available in an egocentric video. Further, in a third person view, the actor's hands are typically occluded or are captured in low resolution, whereas, the hands as well as the objects handled by the actor form an important cue in a first person action recognition setting. Egocentric cameras are usually mounted on a wearer's head and mimic its motion. This introduces a lot of shake in the egocentric video making long term tracking difficult. Availability of labeled data is a general problem in computer vision but is especially acute in egocentric videos since these videos are typically captured in a private context and are unavailable for sharing. This motivates developing novel features and independent analysis for the first person action recognition task.

Our focus is also on doing activity recognition using learned actions grammar.

This task requires a large amount of annotated dataset. Keeping the same in view we introduce a new dataset of an entirely new domain with a large number of videos, actions, objects, and activities. We also test our proposed approach on this data set.

The specific contributions of this thesis are as follows:

1. We posit that motion patterns in the video can be used for recognizing a large variety of first person actions. We conduct the experiments on long and short term actions with and without any handled objects.

2. Using limited samples available for egocentric videos, we show that an LSTM network with RGB and optical flow as input can effectively learn temporal motion patterns of wearer's head, as well as, hands at the same time.

3. The proposed CNN-LSTM framework achieves state of the art performance on various publicly available datasets for first person action recognition.

4. In contrast to most of the contemporary work, we do not use any specific egocentric cues. This makes the proposed approach widely applicable to the newer first person action categories as well.

5. We collect and annotate a new dataset which we name IIITD Plumbing dataset.

6. We present initial results obtained on activity recognition using learned actions' grammar.

1.4 Thesis Outline

The rest of the thesis is organized as follows:

Chapter 2 gives an overview of all the datasets that have been used for experimentation. Experiment methodology is discussed and benchmarks on the newly introduced dataset are reported.

Chapter 3 presents proposed action recognition pipeline and methods.

Chapter 4 reports result obtained using the proposed methods and an analysis of those results.

In Chapter 5, the thesis is concluded and future work is presented.

Chapter 2

Datasets, Experiment Methodology and Benchmarks

2.1 IIITD Plumbing Dataset

We introduce a new egocentric videos dataset of plumbing activities. Videos are captured using a head mounted GoPro camera, a total of six subjects are used. Each subject performs 10 different activities. In total there are 13 different actions and 15 objects. In each action, the subject is handling some object. An activity comprises of a certain number of actions. For two different activities, some actions may be same and the sequence of actions uniquely describe an activity. Dataset consists of 350 videos. for each type of action, we have six different variations due to different color and texture of objects. This has been introduced in the dataset to make sure that the learning method doesn't overfit by learning on specific texture or color. Total time of dataset is 300 minutes. We choose activities such that we have a large number of activities that have actions overlapping in some of the

Figure 2.1: Objects used in the IIITD Plumbing dataset.

activities thus making it usable for grammar learning. This also introduces the increase in the chance of getting an error in action recognition. In the following sections, we discuss the activities being performed, set of actions that make an activity and the objects being handled in each action.

2.1.1 Objects

The IIITD Plumbing data set has following objects; marker, glue, sand paper, hack-saw, pipe, measuring tape, tee joint, elbow joint, nipple joint, union joint, cross joint, gloves, mask and cleaning cloth. The pipes and joints are of different sizes and color to introduce variation in the dataset. The dataset is carefully designed to contain variations throughout out different action and activities but at the same time, there should be an ordered sequence of actions that uniquely creates an ac-

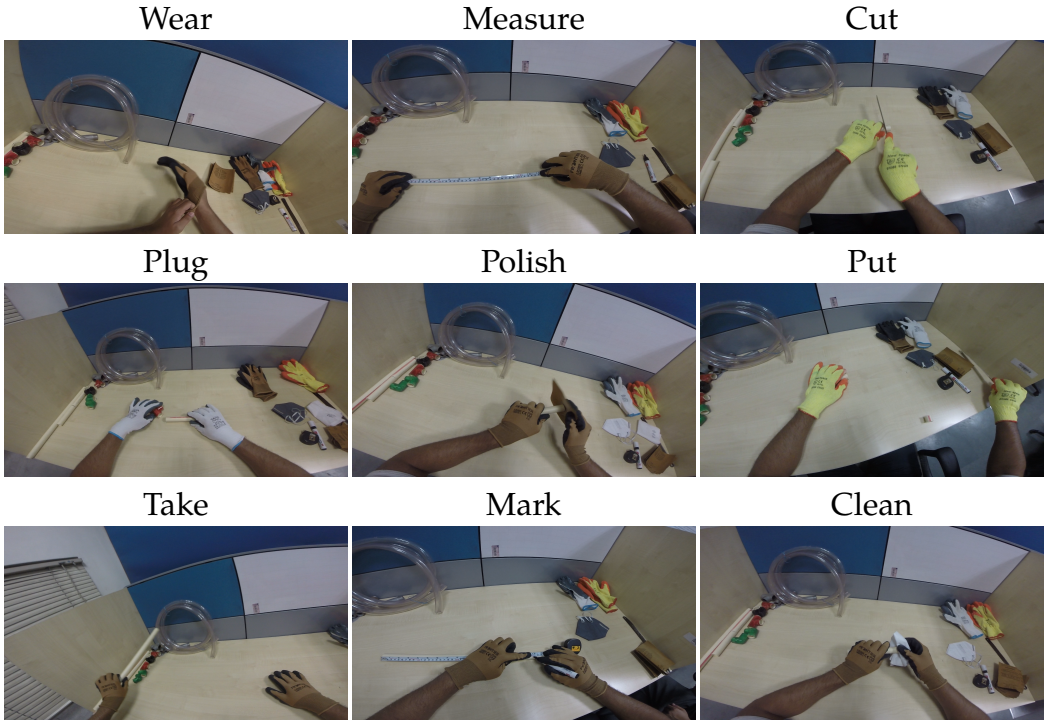


Figure 2.2: Sample actions from the IIITD Plumbing dataset.

tivity.

2.1.2 Actions

1. **Wear:** Action where subject weares either safety goggle, mask or gloves.
2. **Measure:** Subject uses measuring tape to measure pipe.
3. **Mark:** Mark another object like pipe or joint for cutting or applying glue.
4. **Cut:** Subject uses a saw blade to cut down a pipe.
5. **Apply:** Subject applies glue on the pipe.
6. **Plug:** Subject plugs a pipe's end into a joint.

7. **Polish:** Subject polishes a pipe using a sand paper.
8. **Put:** Subject puts down an object on work bench.
9. **Take:** Subject takes an object from the work bench.
10. **Open:** Subject opens a glue bottle.
11. **Close:** Subject closes a glue bottle.
12. **Clean:** Subject cleans a pipe using a cleaning cloth.

2.1.3 Activities

For a given activity we specify the action grammar sequence using the following notation, Action-Object1, and Action-Object1-Object2 for interacting with one object and interacting with two objects respectively.

1. **Wear safety equipments:**
   (a) Take-Gloves
   (b) Wear-Glove
   (c) Wear-Glove
   (d) Take-Goggle
   (e) Wear-Goggle
   (f) Take-Mask
   (g) Wear-Mask.
2. **Measuring a pipe:**

(a) Take-Pipe

(b) Put-Pipe

(c) Take-Measuring Tape

(d) Measure-Pipe

(e) Take-Marker

(f) Mark-Pipe

(g) Put-Measuring Tape

(h) Put-Marker.

3. **Cutting a measured pipe:**

(a) Take-Pipe

(b) Put-Pipe

(c) Take-Saw Blade

(d) Cut-Pipe

(e) Put-Saw Blade.

4. **Polishing a pipe:**

(a) Take-Pipe

(b) Put-Pipe

(c) Take-Sand Paper

(d) Polish-Pipe

(e) Take-Cleaning Cloth

(f) Clean-Pipe

(g) Put-Cleaning Cloth.

5. **Apply glue on polished pipe:**

   (a) Take-Pipe

   (b) Put-Pipe

   (c) Clean-Pipe

   (d) Take-Glue Box

   (e) Open-Glue Box

   (f) Apply-Glue

   (g) Close-Glue Box.

6. **Make tee joint:**

   (a) Take-Pipe1

   (b) Take-Tee Joint

   (c) Plug-Pipe2-Tee Joint End1

   (d) Take-Pipe2

   (e) Plug-Pipe2-Tee Joint End2

   (f) Take-Pipe3

   (g) Plug-Pipe3-Tee Joint End3.

7. **Make Elbow Joint:**

   (a) Take-Pipe1

   (b) Take-Elbow Joint

(c) Plug-Pipe1-Elbow Joint End1

(d) Take-Pipe2

(e) Plug-Pipe2-Elbow Joint End2.

8. **Make union joint:**

(a) Take-Pipe1

(b) Take-Union Joint

(c) Plug-Pipe1-Union Joint End1

(d) Take-Pipe2

(e) Take-Pipe2-Union Joint End2.

9. **Make cross joint:**

(a) Take-Pipe1

(b) Take-Cross Joint

(c) Plug-Pipe1-Cross Joint End1

(d) Take-Pipe2

(e) Plug-Pipe2-Cross Joint End2

(f) Take-Pipe3

(g) Plug-Pipe3-Cross Joint End3

(h) Take-Pipe4

(i) Plug-Pipe4-Cross Joint End4.

10. **Make nipple joint:**

(a) Take-Nipple Joint

(b) Take-Pipe1

(c) Plug-Pipe1-Nipple Joint End1

(d) Take-Pip2

(e) Plug-Pipe2-Nipple Joint End2.

From the above section, it can be seen that the dataset has the property where activities have a fixed action grammar and these actions occur in most of the activities. This dataset reflects a real world scenario where most of the activities have atomic actions that are similar, making it a hard and an interesting problem to solve.

2.2 Other Datasets: GTEA, ADL, UTE, Kitchen, and HUJI

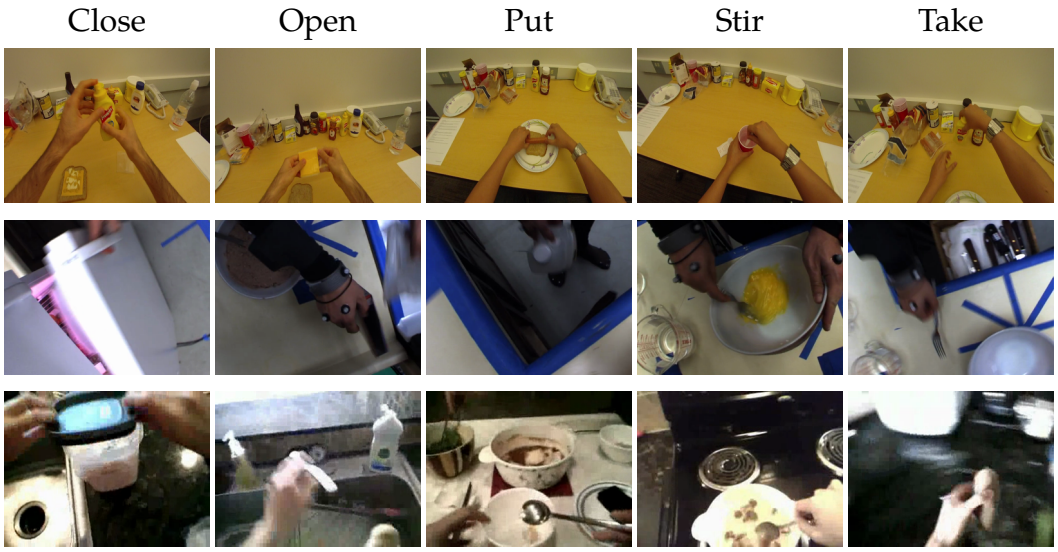


Figure 2.3: Examples of first person action categories where some form of hand-object interactions are present. Top row: GTEA [16], middle row: Kitchen [54], and bottom row: UTE [28].

We have also used other publicly available datasets to test our proposed approach. These datasets are from two action categories. GTEA [16], ADL [39], UTE [28], and Kitchen [54] contains those actions in which some form of hand-object interactions are present. HUJI [42] contains actions in which there are no hand-object interactions. Figure 2.3 shows examples of first person actions where hand-object interaction is present. The figure contains frames from these datasets, GTEA [16] (top row), Kitchen [54] (middle row) and UTE [28] (bottom row). The columns represent the actions 'close', 'open', 'put', 'stir' and 'take'. All the action categories involve manipulating some object but otherwise vary widely across datasets in

terms of appearance and speed of action. The features and technique we suggest in our research are able to successfully recognize the wearer's actions across different presented scenarios, showing the robustness of our method.

Figure 2.4: Examples of first person actions where hand-object interactions are not present. Frames are from HUJI [42] dataset.

For actions involving handled objects, we have used four different publicly available datasets: GTEA [16], Kitchen [54], ADL [39] and UTE [28]. Frame level annotations for the first person actions is available for GTEA [16] and Kitchen [54] datasets. For ADL [39] and UTE [28] datasets we use the annotations provided by Singh *et al.* [51] work, who have annotated a subset of the original dataset where the considered actions occur. Other parts of the video are simply labeled as 'background'. The speed and nature of actions vary across subjects and activities, e.g. action 'open' can mean both 'open' jar or 'open' packet.

The GTEA [16] data set consists of seven long term activities captured using head-mounted cameras. Four subjects have done all the seven activities. Each ac-

tivity lasts for around a minute. We follow 'leave-one-subject-out' experimental setup for all the datasets. In 'leave-one-subject-out', we use the videos of one subject for testing and those of all the other subjects for training. Spatial resolution of a video is 720×405.

The Kitchen [54] data set is captured using a head-mounted camera and IMUs. The spatial resolution of a video is $800 \times 600/1024 \times 768$ captured at 30 FPS. Camera view point is from the top, and jerks in camera motion are quite common. We select seven subjects from the 'Brownie' activity. We use the videos of six subjects for training and test on the videos of the remaining subjects. ADL [39] consists of the videos of subjects performing daily life activities, captured using chest-mounted cameras with 170 degrees of viewing angle. UTE [28] dataset contains 3 to 5 hours of long videos captured from head-mounted cameras taken in the natural and uncontrolled setting.

For actions involving no handled objects we have used HUJI data set [42]. Examples are shown in Figure 2.4. It consists of 14 classes. The dataset is evenly distributed along different classes. Each class consists of many hours of videos containing a total of 82 hours of annotated data. Unlike the action categories are shown in Figure 2.3, there are no hands or handled objects in the scene. However, the motion patterns due to wearer's head form signatures of the first person actions. The columns represent the actions 'Walk', 'Run', 'Stair Climb', 'Stand' and 'Sit' from HUJI data set [42].

Table 2.1 summarizes statistics of various datasets used in our experiments. The proposed approach uses RGB images and dense optical flow and improves the state of the art on all data sets we tested. Results are reported in terms of frame level accuracy, except for HUJI dataset where we use F1 as the accuracy measure.

Dataset	Subjects	Frames	Classes	Accuracy	
				Current	Ours
IIITD Plumbing	6	433924	12	NA	83.72
GTEA [16]	4	31,253	11	68.50	73.45
Kitchen [54]	7	48,117	29	66.23	71.92
ADL [39]	5	93,293	21	37.58	39.43
UTE [28]	2	208,230	21	60.17	65.12
HUJI [42]	NA	1,338,606	14	86	93.92

Table 2.1: Statistics of egocentric video datasets used for experimentation.

The data sets vary widely in appearance, subjects and actions being performed, and the improvement on these data sets validates the generality of the proposed approach for the first person action recognition task.

2.3 Experimental Protocol

2.3.1 Action Recognition

For action category where hand-object interactions are present, we use leave-one-subject-out policy for training and validation and report classification accuracy on the unseen test subjects. We have computed the accuracy in two different ways. Firstly, when we make the prediction for each frame independently. We use a temporal window of W frames around the frame for the prediction.

For real time applications, which involve continuous video understanding, frame level action recognition is important. However, there is a significant confusion with this approach at the action boundaries. To understand the strength of the features, in the second set of experiments, we assume a prior temporal segmentation and predict the action label for a video segment or a splice. In this case, we

predict for each frame in the video segment and then take the majority voting for the label of the segment. However, here we make predictions only for the frames where the temporal window lies completely within the segment. Since there is no confusion at the action boundaries, in this case, the classification accuracy, calculated as the number of frames (or video segments) classified correctly divided by the total number of frames (or video segments), comes out better for the video segments.

For action category where no hand-object interactions are present, i.e. on HUJI [42] dataset, we evaluate the performance in terms of Precision, Recall and F-score to make it comparable with the original work [42].

2.3.2 Activity Recognition

For activity recognition using learned action grammar, we report a total number of videos correctly classified. We are working on a better evaluation strategy that involves per frame classification rather than video classification. The reason this is required is that most of the application involves per frame prediction.

2.4 Benchmarks

2.4.1 Action Recognition Benchmark

For IIITD Plumbing data set we do a benchmark by using the method proposed by Singh *et al.* [51] and Ma *et al.* [33]. This is for the action recognition task. We report the accuracy of different methods and with different features. Features computed are hand motion, head motion, saliency map, and two streams deep learned descriptors. Methods used are 2D and 3D convolution and their combination using

Method	Features	Accuracy
Ego ConvNet(2D)	H	60.19
Ego ConvNet(2D)	H+C	61.47
Ego ConvNet(2D)	H+C+M	63.04
Ego ConvNet(3D)	H	60.07
Ego ConvNet(3D)	H+C	62.17
Ego ConvNet(3D)	H+C+M	62.82
Ego ConvNet(2D)	H+C+M+S+T	73.71
Ego ConvNet(2D)	H+C+M+S+T	74.13
Ego ConvNet(2D)	H+C+M+S+T	74.84

Table 2.2: Benchmark results of different methods presented in [51] on IIITD dataset.

Method	Features	Accuracy
Object CNN	RGB	57.32
Motion CNN	Optical Flow	77.06
Motion and Object CNN joint training	RGB + Optical Flow	80.91

Table 2.3: Benchmark results of different methods presented in [33] on IIITD dataset.

class score fusion. Table 2.2 reports the results obtained. Short hand notations in Table 2.2 are, H: **H**and masks, C: **C**amera/Head motion, M: Saliency **M**ap, S: Deep learned **S**patial descriptors, T: Deep learned **T**emporal descriptors.

Ma *et al.* [33] have presented two networks object detection and motion detection network. They also show that combining these two networks gives the better result. Table 2.3 shows the result obtained for different methods on IIITD dataset.

2.4.2 Activity Recognition Benchmark

We use the method proposed by [26] to benchmark the IIITD plumbing data set for activity recognition task. We follow the same strategy as given in the paper. First Hierarchical Spatial-Temporal Segments are created by extracting body parts and object from video using spectral clustering. To do a temporal clustering space and time distance between pairs of spatial segments are considered. From these fine grained spatial-temporal segments hierarchy is constructed using hierarchical clustering.

Labeling of this segments is done using weakly supervised settings in which some labels are manually assigned to segments and the rest are classified using an SVM trained on Bag of Words of feature space. The accuracy obtained is 68.71. We also try image segmentation using SegNet [9] to compute spatial features. We use weights trained on SUN indoor dataset to classify objects and body parts. The accuracy obtained is 78.23.

Chapter 3

Action and Activity Recognition Algorithms

3.1 Preprocessing

RGB frames Keeping in line with our objective of the end to end training, we use RGB frames directly without any pre-processing (other than scaling to a common size). We first fine tune a CNN model pre-trained on the imagenet dataset. The input for the CNN model is RGB frames of size $N \times M$. We use data augmentation approach at this stage to increase the data. We crop $N' \times M'$ sized image from this input at various locations and feed it to the network. This increases the amount of data. We predict an action class for each frame in the video. However, to include temporal information in the prediction process, we pick a temporal window of size w around each frame to be given as input. Note that an action itself may be longer than the w frames. In this case, the temporal window approach may also be seen as data augmentation in the temporal domain.

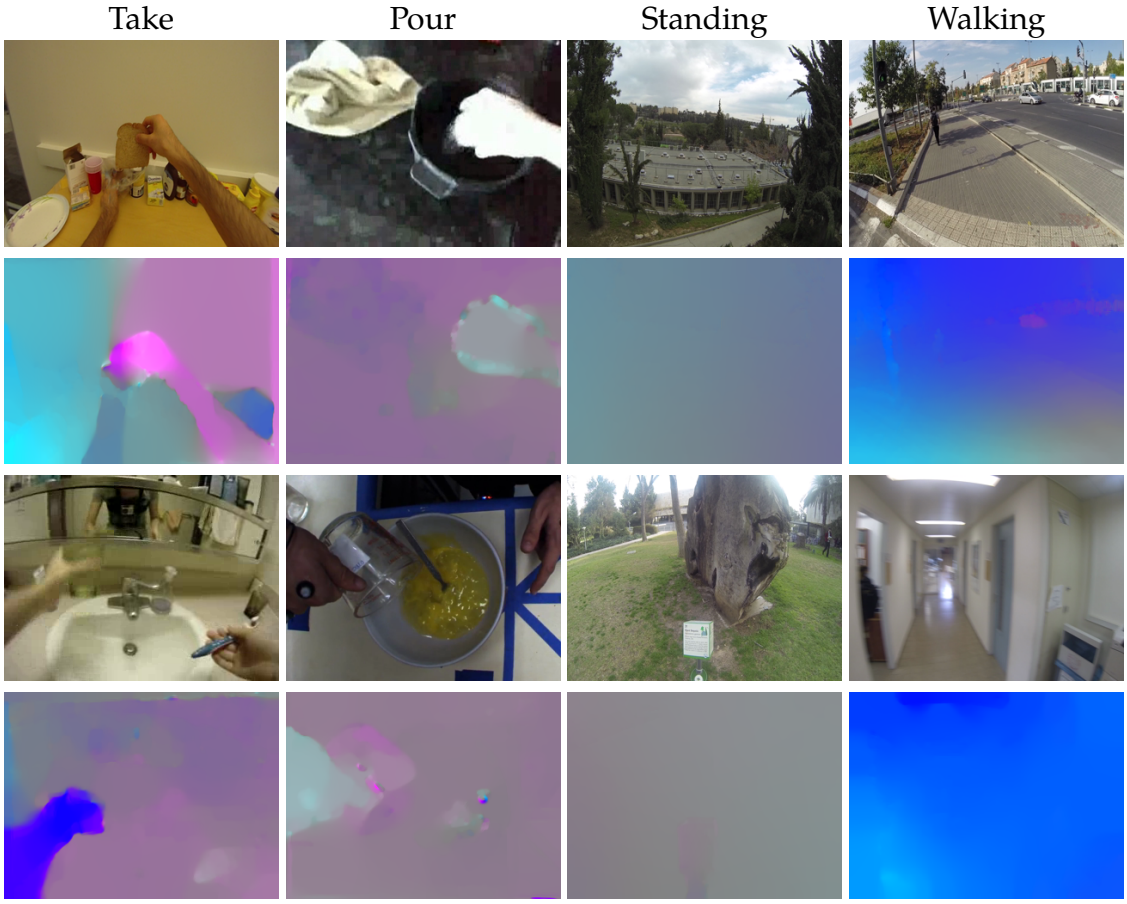


Figure 3.1: Sample RGB frames and their corresponding optical flows from different datasets showing variation in dataset.

Optical flow Motion patterns, as indicated by optical flow in an image are crucial indicators of first person actions in the proposed architecture. It may be noted that, unlike some of the earlier works [41, 42] using sparse optical flow, we propose to use dense optical flow [11], as some of the fine object manipulation activities are hard to capture in a sparse scenario. Further, unlike the prior work using trajectories as the cue for motion saliency [50, 51], we make no attempt to track any feature point or evaluate motion saliency in any other way. However, movement

of wearer's head is often the dominant source of flow in the frame and may be unrelated to the action being performed. Therefore, as suggested in some of the earlier works [50, 51], we compensate the flow due to head movement by canceling frame to frame homography. We use this compensated flow as input to our network.

Preprocessing Due to different evaluation methodology and huge size mismatch of datasets, we have used them slightly differently in our experiments. For actions where we have hands and object interaction we use 'leave-one-subject' policy and create a training set from 3 subjects and test on the fourth subject. This is in consonance with other contemporary works [13, 51] and makes us comparable with them. The size of datasets for this type of action category is very small as compared to other so we use a stride of 6 for taking samples from these datasets. For long term action categories, we keep a stride of 11 for choosing samples. We randomly select 650 sequences for the training set and rest of the sequences are used for testing.

3.2 Action Recognition

CNN We use a convolutional neural network to learn visual features from video frames. To make up for the scarcity of training data, we have used pre-trained CNN models and fine tuned them on our datasets. We have compared using two different CNN models based upon their proven performance on other computer vision problems:

1. Hybrid-CaffeNet [66]: The CNN model consists of five convolutional layers. The first and second convolutional layers are followed by a MAX pooling

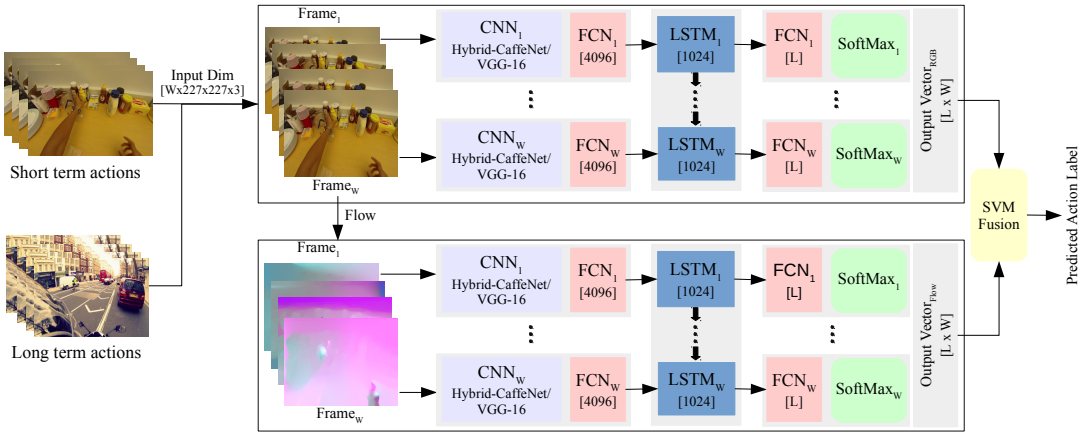


Figure 3.2: CNN-LSTM achitecture for action recognition.

layer, a ReLU non-linear activation layer, a local response normalization layer. The third and fourth convolutional layers only have ReLU activation layers. The fifth convolution layer is followed by a MAX pooling layer, a RELU layer, and three fully-connected layers. We use SOFTMAX activation for calculating class probabilities and multinomial logistic loss during training. We take the output of the second last fully-connected layer and give it as an input to the LSTM model.

2. VGG-16 [49]: The VGG-16 network consists of 15 convolutional layers and two fully-connected layers. Every convolutional layer is followed by a MAX pooling layer. VGG-16 network has been trained on a large dataset and is very deep. We fine tune only the last two fully-connected layers of VGG-16 on our datasets. We have used the output of the second last fully-connected layer for input to the LSTM model.

CNN training We use Hybrid-CaffeNet's pre-trained model trained on the imagenet dataset. We fine-tune third, fourth and fifth CNN layers and last three FC layers. To fine-tune on RGB images we use a batch size of 128 images. Images are resized to $(320 \times 240 \times 3)$ pixels. The input to the network is formed by $(227 \times 227 \times 3)$ RGB frames, cropped randomly from the resized images. Input frames are normalized by subtracting mean pixel values from each pixel of the input frame. Mean pixel value is computed over the whole data set. Learning rate of 0.001, learning momentum of 0.9, gamma of 0.1 and weight decay of 0.005 are used. Learning rate is decreased by one-tenth after every 10K iterations. The model is trained for 50K iterations. For optical flow, we use the same batch size. Weight decay, gamma, learning rate and momentum are similar to those of the RGB model. We choose a step-size of 20K and perform training for 70K iterations.

LSTM One layer of LSTM units have been used with each LSTM having C cells. The number of units is equal to the number of frames W given as input, and are connected in a unidirectional fashion in the layer. The input to the LSTM network is the output of the CNN network. We use shared weights for the CNN part in the CNN-LSTM model. This keeps network small as only one CNN is shared with multiple LSTM units. The output of CNN-LSTM depends on the number of input frames given to it. If W frames are given as an input, then W class labels form the output, each representing the class for a particular frame.

Fusion We use two different CNN-LSTM models, one for RGB and one for flow. The input to these models is a temporal window of W frames. They give W vectors as output, each vector is of length L, where L is the number of classes in the dataset. It has been shown in the earlier works that combining LSTM outputs in

various ways (first/ last/ average/ max-frequency) does not affect the accuracy in any significant way. In the proposed model we make $L \times W$ dimensional feature map from each stream and fuse it with a 'RBF-kernel' SVM.

LSTM training We train two streams, one for RGB frames and one for optical flow. We use the fine-tuned Hybrid-CaffeNet network's weights for the CNN part of the CNN-LSTM network. We take the output of the second fully-connected layer and give it to an LSTM unit with 1024 cells. There are 11 LSTM units, corresponding to 11 input frames, connected in a unidirectional manner. The base learning rate is kept at 0.001, decreased by one-tenth after every 10K iterations. The momentum of 0.9, gamma of 0.1 and weight decay of 0.005 are used. We train the model for 50K iterations. For training on flow frames, we keep the momentum, gamma, weight decay and base learning rate same as those of RGB model. Learning rate is decreased by one-tenth after every 20K iterations and training is stopped at 70K iterations.

Varying Network Architecture We have experimented with multiple variations in the proposed model. We have tried Hybrid-CaffeNet and VGG-16 for the CNN part. We have explored different approaches for fine tuning Hybrid-CaffeNet, the first approach is to fine tune only FC layers and second is to fine tune all but first two CNN layers. In VGG-16 we fine tune the second last FC layer. We have also experimented with different values of C (number of LSTM cells) as 128, 256, 512, 1024 and 2048 and found 1024 cells to be the best. Similarly, we have tested with various temporal window sizes (5, 11, 15, 21 frames) and found 11 to be a good compromise between accuracy and size of the network.

All experiments have been conducted on a machine with Xeon E5 CPU with

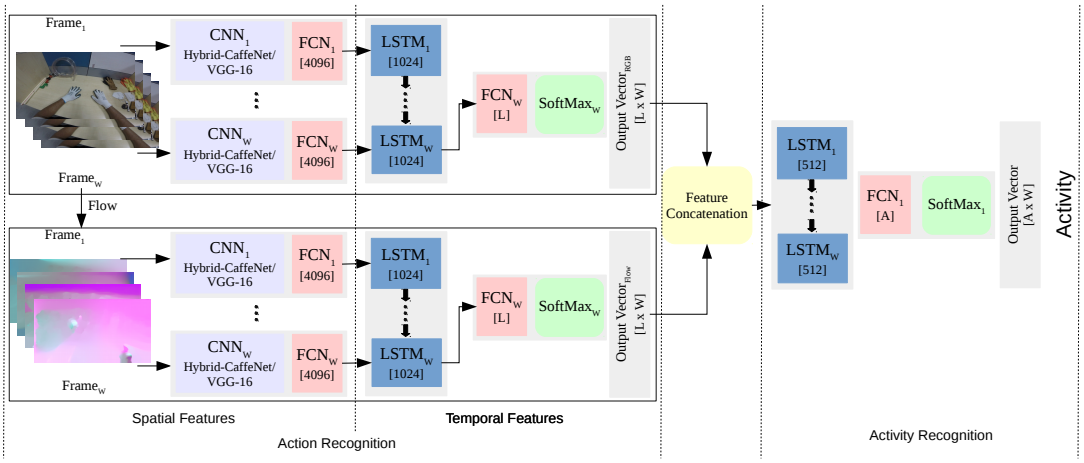


Figure 3.3: Proposed achitecture for activity recognition.

4 cores, 16 GB RAM containing single *Nvidia Titan X* GPU and running Ubuntu 15.10. We use Caffe [20] and tensor-flow [7] deep learning libraries for all our experiments.

3.3 Activity Recognition

For activity recognition, we take the output of the LSTMs from the two streams of proposed action recognition CNN-LSTM architecture, concatenate them and give it as input to another layer of LSTM. For this network, the output label becomes the activity being performed in the video. We fix the LSTM cells to 512 and use a single layer LSTM. Proposed architecture is shown in Figure 3.3. Learning rate of 0.001, learning momentum of 0.9, gamma of 0.1 and weight decay of 0.005 are used. Learning rate is fixed for whole training. The model is trained for 10K iterations.

Chapter 4

Experimental Results

4.1 Action Recongition

Actions with handled object This action category consists of short term actions where wearer handles some objects. We perform experiments on all five datasets described in Chapter 2. We follow leave-one-subject-out experimental setup for all the experiments.

Table 4.1 analyzes the impact of using optical flow in our experiments. Unlike [35], who have reported higher performance using RGB features and inferior

Input	**Frame level Accuracy**				
	IIITD Plumbing	**GTEA** [16]	**Kitchen** [54]	**ADL** [39]	**UTE** [28]
RGB	78.91	66.31	62.23	35.02	59.10
Flow	80.24	72.03	69.90	38.43	64.78
Combined	83.72	73.45	71.92	39.43	65.12

Table 4.1: Analysis of the proposed model using only RGB, flow and combined input.

Window Size	RGB	Flow	Combined
5	64.14	67.01	68.56
11	66.31	72.03	73.45
15	66.47	71.94	73.97
21	67.49	72.11	74.37

Table 4.2: Frame level classification results for different temporal window size.

Architecture	RGB	Flow	Class score fusion
Hybrid-CaffeNet	66.31	72.03	73.45
VGG-16	66.29	72.17	73.94

Table 4.3: Frame level classification results for two architectures.

performance using optical, we observe the opposite. we speculate this could be attributed to using raw optical flow in [35]. In our experiments also using raw optical flow gives a much much inferior performance at 49% compared to 72% using compensated flow. Figure 4.1 shows the activations of first convolution layer for RGB and flow frames of two different classes. We also note that a recent paper by Ma *et al.* [33] have reported an accuracy of 75.08% for the task of first person action classification on the GTEA dataset. However, it is not clear from their description if they have included the 'background' class in their accuracy evaluation. We observe an accuracy of 84% for the action classification if we do not include background class in the test set.

Table 4.2 shows results for different temporal window size. For the window size of 5, the network performs poor as compared to the window size of 11. When temporal size is increased to 11, there is a sudden change in network performance this is because 5 frames are not enough to capture motion patterns in an action. Any further increase in temporal window size only results in a slightly better per-

Dataset	Accuracy		
	Frame level	**Segment level**	**Chance level**
IIITD Plumbing	83.72	89.21	7.3
GTEA [16]	73.45	82.91	11
Kitchen [54]	71.92	74.19	3.4
ADL [39]	39.43	42.07	4.7
UTE [28]	65.12	69.73	4.7

Table 4.4: Frame level and segment level results.

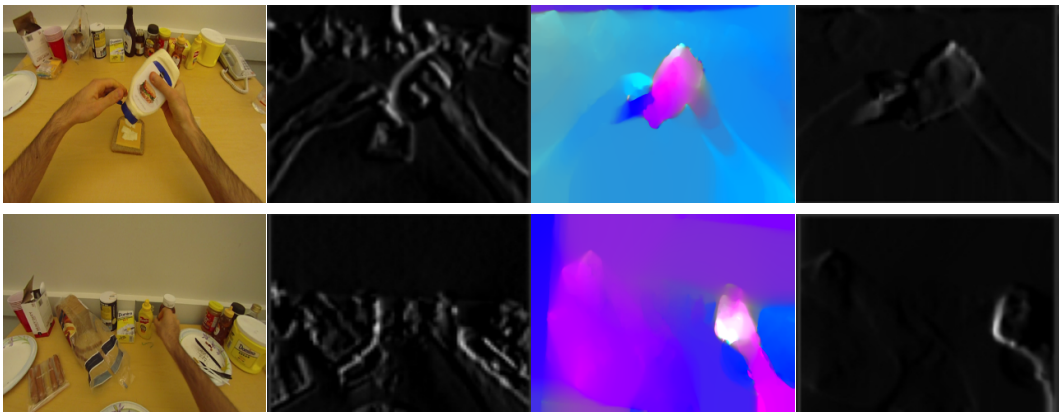


Figure 4.1: Activations of first convolutional layer.

formance. Note that, most actions in this category are very short with the number of frames under 50. We note that [35] reports an accuracy of 65% using the single frame and 67% using LSTM indicating that network is probably not able to adequately exploit the sequence structure. On the other hand, we observe an accuracy of 29% using single frame optical flow and 72% using LSTM. For RGB frame, we achieve an accuracy of 43% using the single frame and 66% using video sequence. The numbers indicate that the proposed model is able to successfully exploit the sequence structure in first person actions.

We do not observe any significant difference in performance using different

Figure 4.2: Class probabilities for correct and incorrect prediction.

types of CNN models. Table 4.3 suggests that Hybrid-CaffeNet and VGG-16 CNN-LSTM models perform similarly. The difference in their performances can be attributed to chance and doesn't suggest that one type of network is better than other.

Table 4.4 shows frame level and segment level results on all five datasets. Segment level gives better results than the frame level due to clear boundaries. In the case of frame level, at boundaries, the network gets confused due to different motion patterns across the boundary.

We have experimented with our method on other publicly available egocentric datasets as well. Table 2.1 shows results obtained by our model and previous works on all five datasets. Singh *et al.* [51] reports an accuracy of 68.50 on GTEA [16] dataset, 66.23 on Kitchen [54] dataset, 37.58 on ADL [39] dataset and 60.17 on UTE dataset. We achieve 73.45 on GTEA [16] dataset, 71.92 on Kitchen [54] dataset, 39.43 on ADL [39] dataset and 65.12 on UTE [28] dataset. The experiments show that the proposed approach consistently outperforms state of the art accuracy for each of the datasets.

Figure 4.2 analyzes some of the failure cases of GTEA [16] dataset. We see large similarities between some of the first person actions and see most of the errors concentrated there. We believe some of the recent works in fine grained classification task may be useful here and will be the focus of our attention in the future.

Actions without handled object This action category consists of long term actions and all the classes except 'cooking' does not involve any object manipulation. We follow the evaluation strategy of [42] and get an average recall rate of 93% against the 86% reported by them. Table 4.5 summarizes the results. We believe that our CNN-LSTM network is able to perform better due to RGB frames as the feature. Their network confuses in actions where the flow is often ambiguous like

'sitting' and 'standing'. Our RGB model learns features complementary to the flow based model.

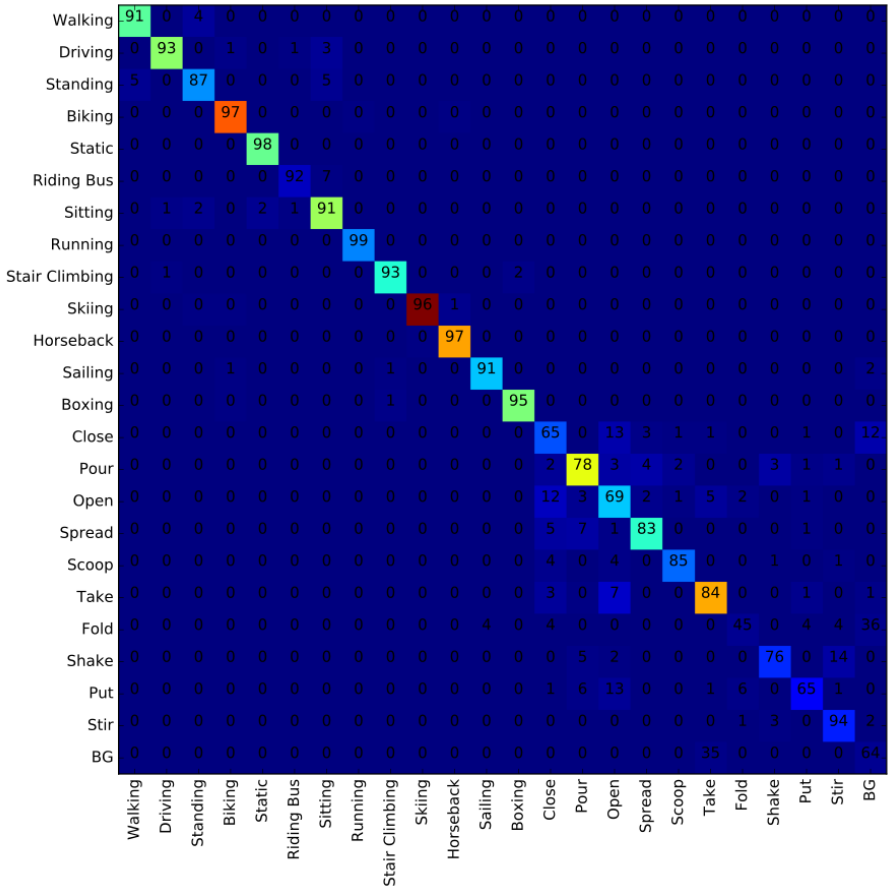


Figure 4.3: Confusion matrix showing action recognition results obtained when GTEA [16] and HUJI [42] dataset are used.

Mixed Action Classification To the best of our knowledge, none of the state of the art can classify first person actions with and without a handled object. In a real life setting, it is much more likely for a wearer to be involved in a mixed action setting. We test our model for such a scenario by mixing the samples from GTEA

and HUJI datasets. Figure 4.3 gives the confusion matrix for the experiments. It is evident that the proposed network does not seem to have any confusion in the two different category of actions and the confusing pairs remain the usual similar actions such as shake and stir only. The experiment indicates much wider and practical applicability of the proposed technique compared to state of the art.

4.2 Activity Recognition

Due to the limited number of activity videos in GTEA, ADL, UTE and Kitchen datasets we only use IIITD Plumbing data set for Activity Recognition. Our method is not able to learn on small datasets. On IIITD Plumbing dataset we use a leave-one-subject out policy for testing. The proposed method achieves an accuracy of 94.31% on activity recognition task using learned actions' grammar. These results are obtained in the primitive phase of network design and experimentation.

Class	Precision		Recall		F1-Score
	[42]	Ours	[42]	Ours	
Walking	0.93	0.91	0.91	0.95	0.93
Driving	0.94	0.93	0.98	0.95	0.94
Standing	0.62	0.87	0.59	0.83	0.85
Biking	0.92	0.97	0.94	0.96	0.97
Static	0.44	0.98	0.99	0.97	0.98
Riding Bus	0.94	0.92	0.87	0.80	0.85
Sitting	0.73	0.91	0.71	0.91	0.91
Running	0.91	0.99	0.78	0.96	0.98
Stair Climbing	1.00	0.93	0.59	0.95	0.94
Skiing	0.92	0.96	0.82	0.99	0.98
Horseback	1.00	0.97	0.92	0.96	0.97
Sailing	0.65	0.91	0.99	0.96	0.96
Boxing	0.47	0.95	0.93	0.97	0.96
Cooking	0.71	0.94	0.89	0.98	0.96
Plumbing	0.69	0.93	0.86	0.97	0.96
Mean	**0.79**	**0.94**	**0.87**	**0.93**	**0.94**

Table 4.5: Results for the action categories when there is no interaction between wearer's hands and object.

Chapter 5

Conclusion and Future Work

Earlier works for first person action recognition have explored various egocentric cues such as the motion of wearers head or hands and objects present in the scene. We propose in this thesis that motion patterns in an egocentric video alone are sufficient to recognize all kinds of first person actions. We show that a CNN-LSTM network is sufficient to learn such motion patterns and can recognize first person actions, both with or without a handled object. This greatly simplifies the design, allowing wider applicability of the proposed framework. The work on activity recognition using learned actions' grammar is in its primitive stage.

Future work in activity recognition is to decrease the number of parameters in the whole network. We would also like to get a per frame activity label and action label using a single network. We hope that using Bi-LSTM network for grammar learning can prove an important point that an activity can uniquely be determined by the sequence of actions.

Bibliography

[1] Google glass. https://www.google.com/glass/start/.

[2] Gopro. http://gopro.com/.

[3] Microsoft hololens. http://research.microsoft.com/en-us/um/cambridge/projects/sensecam/.

[4] Microsoft sensecam. https://www.microsoft.com/microsoft-hololens/en-us.

[5] Object recognition. http://umair-khan.quest.edu.pk/qa.

[6] Pivothead. http://www.pivothead.com/.

[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. 2016.

[8] O. Aghazadeh, J. Sullivan, and S. Carlsson. Novelty detection from an ego-centric perspective. In *CVPR*, 2011.

[9] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[10] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara. Gesture recognition in ego-centric videos using dense trajectories and hand segmentation. In *CVPRW*, 2014.

[11] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36. Springer, May 2004.

[12] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[13] A. Fathi, A. Farhadi, and J. M. Rehg. Understanding egocentric activities. In *ICCV*, 2011.

[14] A. Fathi, J. K. Hodgins, and J. M. Rehg. Social interactions: A first-person perspective. In *CVPR*, 2012.

[15] A. Fathi, Y. Li, and J. M. Rehg. Learning to recognize daily actions using gaze. In *ECCV*, 2012.

[16] A. Fathi, X. Ren, and J. M. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011.

[17] Y. Hoshen and S. Peleg. Egocentric video biometrics. *CoRR, abs/1411.7591*, 2014.

[18] M. Jain, H. Jégou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013.

[19] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. In *TPAMI*, 2013.

[20] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, , and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *arXiv preprint arXiv:1408.5093*, 2014.

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[22] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast unsupervised ego-action learning for first-person sports videos. In *CVPR*, 2011.

[23] A. Klaser and M. Marszalek. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[24] E. Kraft and T. Brox. Motion based foreground detection and poselet motion features for action recognition. In *ACCV*, 2014.

[25] T. Lan, L. Sigal, and G. Mori. Social roles in hierarchical models for human activity recognition. In *CVPR*, 2012.

[26] T. Lan, Y. Zhu, A. R. Zamir, and S. Savarese. Action recognition by hierarchical mid-level action elements. 2015.

[27] I. Laptev. On space-time interest points. *IJCV*, 2005.

[28] Y. J. Lee, J. Ghosh, and K. Grauman. Discovering important people and objects for egocentric video summarization. In *CVPR*, 2012.

[29] C. Li and K. M. Kitani. Pixel-level hand detection in ego-centric videos. In *CVPR*, 2013.

[30] Y. Li, A. Fathi, and J. M. Rehg. Learning to predict gaze in egocentric video. In *ICCV*, 2013.

[31] Y. Liu, L. Nie, L. Liu, and D. S. Rosenblum. From action to activity: Sensor-based activity recognition. *Neurocomputing*, 181:108 – 115, 2016.

[32] Z. Lu and K. Grauman. Story-driven summarization for egocentric video. In *CVPR*, 2013.

[33] M. Ma, H. Fan, and K. M. Kitani. Going deeper into first-person activity recognition. In *CVPR*, 2016.

[34] K. Matsuo, K. Yamada, S. Ueno, and S. Naito. An attention-based activity recognition for egocentric video. In *CVPRW*, 2014.

[35] J. Y. Ng, M. J. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *CoRR*, abs/1503.08909, 2015.

[36] F. Niu and M. Abdel-Mottaleb. Hmm-based segmentation and recognition of human activities from video sequences. In *2005 IEEE International Conference on Multimedia and Expo*, pages 804–807, 2005.

[37] K. Ogaki, K. M. Kitani, Y. Sugano, and Y. Sato. Coupling eye-motion and ego-motion features for first-person activity recognition. In *CVPRW*, 2012.

[38] H.-K. Peng, P. Wu, J. Zhu, and J. Y. Zhang. Helix: Unsupervised grammar induction for structured activity recognition. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, 2011.

[39] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012.

[40] Y. Poleg, C. Arora, and S. Peleg. Head motion signatures from egocentric videos. In *ACCV*, 2014.

[41] Y. Poleg, C. Arora, and S. Peleg. Temporal segmentation of egocentric videos. In *CVPR*, 2014.

[42] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora. Compact cnn for indexing egocentric videos. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016.

[43] X. Ren and C. Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *CVPR*, 2010.

[44] X. Ren and M. Philipose. Egocentric recognition of handled objects: Benchmark and analysis. In *CVPRW*, 2009.

[45] N. Rhinehart and K. M. Kitani. Learning action maps of large environments via first-person vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[46] M. S. Ryoo and L. Matthies. First-person activity recognition: What are they doing to me? In *CVPR*, 2013.

[47] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACMMM*, 2007.

[48] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[49] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[50] S. Singh, C. Arora, and C. V. Jawahar. Trajectory aligned features for first person action recognition. *Pattern Recognition*, 2016.

[51] S. Singh, C. Arora, and C. V, Jawahar. First person action recognition using deep learned descriptors. In *CVPR*, 2016.

[52] H. Soo Park, J.-J. Hwang, Y. Niu, and J. Shi. Egocentric future localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[53] H. Soo Park, j.-J. Hwang, and J. Shi. Force from motion: Decoding physical sensation in a first person video. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[54] E. H. Spriggs, F. De La Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *CVPRW*, 2009.

[55] S. Sundaram and W. W. M. Cuevas. High level activity recognition using low resolution wearable vision. In *CVPRW*, 2009.

[56] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, 2010.

[57] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[58] T. L. M. van Kasteren, G. Englebienne, and B. J. A. Kröse. Hierarchical activity recognition using automatically clustered actions. In *Proceedings of the Second International Conference on Ambient Intelligence*, AmI'11, 2011.

[59] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

[60] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[61] J. Wang, Y. Cheng, and R. Schmidt Feris. Walk and learn: Facial attribute representation learning from egocentric video and contextual data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[62] G. Willems, T. Tuytelaars, and L. Van Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.

[63] B. Xiong, G. Kim, and L. Sigal. Storyling representation of egocentric videos with an application to story-based search. In *ICCV*, 2015.

[64] L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *ICCV*, 2009.

[65] R. Yonetani, K. M. Kitani, and Y. Sato. Recognizing micro-actions and reactions from paired egocentric videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[66] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[67] Y. Zhou, B. Ni, R. Hong, X. Yang, and Q. Tian. Cascaded interactional targeting network for egocentric video analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.