UNIVERSITÀ DEGLI STUDI
DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
ICT International Doctoral School

DEEP NEURAL ARCHITECTURES FOR VIDEO REPRESENTATION LEARNING

Swathikiran Sudhakaran

Advisor

Dr. Oswald Lanz

Fondazione Bruno Kessler, Trento, Italy

June 2019

To my parents, Prasanna and Sudhakaran

Abstract

Automated analysis of videos for content understanding is one of the most challenging and well researched areas in computer vision and multimedia. This thesis addresses the problem of video content understanding in the context of action recognition. The major challenge faced by this research problem is the variations of the spatio-temporal patterns that constitute each action category and the difficulty in generating a succinct representation encapsulating these patterns. This thesis considers two important aspects of videos for addressing this problem: (1) a video is a sequence of images with an inherent temporal dependency that defines the actual pattern to be recognized; (2) not all spatial regions of the video frame are equally important for discriminating one action category from another. The first aspect shows the importance of aggregating frame level features in a sequential manner while the second aspect signifies the importance of selective encoding of frame level features. The first problem is addressed by analyzing popular Convolutional Neural Network (CNN)-Recurrent Neural Network (RNN) architectures for video representation generation and concludes that Convolutional Long Short-Term Memory (ConvLSTM), a variant of the popular Long Short-Term Memory (LSTM) RNN unit, is suitable for encoding spatio-temporal patterns occurring in a video sequence. The second problem is tackled by developing a spatial attention mechanism for the selective encoding of spatial features by weighting spatial regions in the feature tensor that are relevant for identifying the action category. Detailed experimental analysis carried out on two video recognition tasks showed that spatially selective encoding is indeed beneficial. Inspired from the two aforementioned findings, a new recurrent neural unit, called Long Short-Term Attention (LSTA), is developed by augmenting LSTM with built-in spatial attention and a revised output gating. The first enables LSTA to attend to the relevant spatial regions while maintaining a smooth tracking of the attended regions and the latter allows the network to propagate a filtered version of the memory localized on the most discriminative components of the video. LSTA surpasses the recognition accuracy of existing state-of-the-art techniques on popular egocentric activity recognition benchmarks, showing its effectiveness in video representation generation.

Keywords
[Computer Vision, Action Recognition, Spatial Attention, Recurrent Neural Networks, Video Representation Learning]

Acknowledgements

During the course of this PhD, I had the privilege to learn from, work with and enjoy the company of many brilliant people who played a significant role in the completion of this PhD study.

Thanks to my supervisor Dr. Oswald Lanz for his valuable advices and suggestions that helped me in shaping up my research and the immense patience that he had with me. He was always available for discussions and brainstorming, which was instrumental in reaching at this stage.

Thanks to Prof. Sergio Escalera for providing me the opportunity to spend three months as a visiting researcher in the Human Pose Recovery and Behavior Analysis (HuPBA) group at the Computer Vision Center (CVC), Barcelona, which has allowed me in furthering my knowledge of the research area.

Thanks to Prof. Alex James, my Master's thesis advisor, from whom I learned how to do research and write a paper and for advising me to pursue a PhD study.

Thanks to my colleagues in the TeV lab, FBK, for all their help and presence that has allowed me in the completion of this PhD; especially Dr. Paul Ian Chippendale and Dr. Fabio Poiesi for the enjoyable conversations during the lunch time and for planning the outings and Via Ferratas; Massimiliano Mancini, Andrea Simonelli and Alessio Xompero for making the time spent in the lab delightful and for the dinners and pizza evenings.

Thanks to my friends Sudipan Saha and Dr. Alexander Vostroknutov who helped me in having a social life, outside research, and making it more enjoyable and fun.

Finally, thanks to my parents and my sister who supported me throughout my academic life and for always encouraging me to pursue education. None of this would have been possible without them.

Contents

1 Introduction **1**

1.1 Objective 2

1.2 Overview of the Thesis 4

1.3 Contributions 6

2 Recurrent Convolutional Encoding for Video Representation Learning **7**

2.1 CNN-RNN Architecture for Video Representation Learning 8

2.2 Convolutional Long Short-Term Memory 10

2.3 CNN-RNN for Detecting Violent Videos 11

2.3.1 Related Works 12

2.3.2 Proposed Method 14

2.3.3 Experiments and Results 16

2.4 CNN-RNN for First Person Interaction Recognition from Videos 21

2.4.1 Related Works 22

2.4.2 Proposed Method 23

2.4.3 Experiments and Results 26

2.5 Conclusion 33

3 Spatially Selective Encoding for Video Representation Learning **35**

3.1 Class Activation Mapping 36
3.2 Spatial Attention for Egocentric Activity Recognition . . . 38
3.2.1 Related Works 40
3.2.2 Proposed Method 42
3.2.3 Experiments and Results 44
3.3 Spatial Attention for Third Person Action Recognition . . 54
3.3.1 Related Works 55
3.3.2 Proposed Method 57
3.3.3 Experiments and Results 62
3.4 Conclusion 70

4 Recurrent Convolutional Spatially Selective Encoding for Video Representation Learning 71
4.1 Analysis of LSTM 73
4.2 Long Short-Term Attention 75
4.2.1 Attention Pooling 75
4.2.2 Output Pooling 78
4.3 LSTA for Egocentric Activity Recognition 79
4.3.1 Related Works 80
4.3.2 Proposed Method 83
4.3.3 Experiments and Results 86
4.4 Conclusion 99

5 Conclusions and Future Work 101
5.1 Conclusions 101
5.2 Future Work 102

Bibliography 105

List of Tables

2.1 Classification accuracy obtained with the hockey fight dataset for different models 18
2.2 Comparison with state-of-the-art techniques for violence recognition 19
2.3 Comparison between ConvLSTM and LSTM models 21
2.4 Comparison with state-of-the-art methods for interaction recognition 30
2.5 Comparison with methods that use RGB-D images 33

3.1 Ablation analysis on the fixed split of GTEA 61 dataset . . 46
3.2 Comparison with state-of-the-art methods 49
3.3 Comparison of recognition accuracy on EGTEA Gaze+ dataset 50
3.4 Comparison of Top-down Attention Recurrent VLAD Encoder (TA-VLAD) against state-of-the-art methods on HMDB51 and UCF101 datasets 69

4.1 Ablation analysis on GTEA 61 fixed split 87
4.2 Detailed ablation analysis on GTEA 61 fixed split 87
4.3 Comparative analysis on GTEA 61 fixed split 94
4.4 Comparison of LSTA with state-of-the-art methods on popular egocentric datasets 98
4.5 Comparison of LSTA with state-of-the-art methods on EPIC-KITCHENS dataset 98

List of Figures

1.1 Overview of the thesis 4

2.1 CNN-RNN architecture for video representation generation 9
2.2 Block diagram of the violence recognition model. 14
2.3 Block diagram of the first person interaction recognition network 24

3.1 Class Activation Map (CAM) obtained for frames from GTEA 61 dataset using ResNet-34 trained on ImageNet 37
3.2 Block diagram of the egocentric activity recognition model 42
3.3 t-SNE embedding of the features 47
3.4 Confusion matrix of split P1 from GTEA Gaze+ dataset . 51
3.5 Confusion matrix of split P3 from GTEA Gaze+ dataset . 52
3.6 Spatial attention maps obtained for frames from GTEA 61 dataset 53
3.7 Block diagram of TA-VLAD architecture 57
3.8 Illustration of VLAD encoding and spatial attention map generation methods 60
3.9 Results of ablation analysis 67
3.10 Sample Class Activation Maps on frames from HMDB51 dataset 68

4.1 Illustration of LSTA 76
4.2 Illustration of the cross-modal fusion approach 83

4.3 Improved categories by adding output pooling to the baseline 89
4.4 Improved categories by adding attention pooling to the baseline 90
4.5 Improved categories by adding pooling to the baseline . . . 91
4.6 Improved categories by using LSTA compared to the baseline 92
4.7 Class categories improved by using cross-modal fusion over two stream late fusion 93
4.8 Class categories improved by using LSTA over eleGAtt . . 95
4.9 Class categories improved by using LSTA over ego-rnn . . 96
4.10 Sample attention maps generated on the fixed split of GTEA 61 dataset 97

List of Abbreviations

BN Batch Normalization

CAM Class Activation Map

ConvLSTM Convolutional Long Short-Term Memory

CNN Convolutional Neural Network

FC Fully Connected

GRU Gated Recurrent Unit

HOF Histogram of Optical Flow

LRCN Long-Term Recurrent Convolutional Neural Network

LSTA Long Short-Term Attention

LSTM Long Short-Term Memory

MBH Motion Boundary Histogram

MoIWLD Motion Improved Weber Local Descriptor

MoSIFT Motion Scale-Invariant Feature Transform

MoWLD Motion Weber Local Descriptor

MPEG Moving Picture Experts Group

NN Neural Network

OViF Oriented Violent Flows

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

SGD Stochastic Gradient Descent

STIP Space-Time Interest Points

TA-VLAD Top-down Attention Recurrent VLAD Encoder

TSN Temporal Segment Network

ViF Violent Flows

VLAD Vector of Locally Aggregated Descriptors

Publications

[Pub1] Swathikiran Sudhakaran and Oswald Lanz. Learning to Detect Violent Videos using Convolutional Long Short-Term Memory. In *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017.

[Pub2] Swathikiran Sudhakaran and Oswald Lanz. Convolutional Long Short-Term Memory Networks for Recognizing First Person Interactions. In *Proc. ICCV Workshops*, 2017.

[Pub3] Swathikiran Sudhakaran and Oswald Lanz. Attention is All We Need: Nailing Down Object-centric Attention for Egocentric Activity Recognition. In *Proc. BMVC*, 2018.

[Pub4] Swathikiran Sudhakaran and Oswald Lanz. Top-down Attention Recurrent VLAD Encoding for Action Recognition in Videos. In *Proc. 17th International Conference of the Italian Association for Artificial Intelligence*, 2018.

[Pub5] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. LSTA: Long Short-Term Attention for Egocentric Action Recognition. In *Proc. CVPR*, 2019.

[Pub6] Swathikiran Sudhakaran and Oswald Lanz. Top-down Attention Recurrent VLAD Encoding for Action Recognition in Videos. *Intelligenza Artificiale*, 2019 (minor revision pending).

Chapter 1

Introduction

Recently, with the development of affordable cameras and smart phones, the number of videos getting uploaded to video streaming websites and social media has profoundly increased. This along with the recent widespread deployment of cameras for surveillance necessitates the development of automated systems for analyzing and understanding the content of videos. The applications of such automated systems for video content analysis ranges from surveillance to behavior understanding, video indexing and retrieval, human-computer interaction or human assistance, to name a few. This video content understanding problem has been well researched in computer vision. However, the performance of such automated systems still have not reached to a level that enables them to be applied in a practical use-case scenario. Action recognition from videos is a subset of the video content understanding problem. Development of techniques for generating video representations suitable for action classification is a stepping stone to resolving the more general problem of video content analysis.

Recent advances in deep learning highly benefited problems such as image classification [1, 2] and object detection [3, 4]. However, the performance of deep learning methods for action recognition from videos is still not comparable to the advances made in object recognition from still

images [1]. One of the main difficulties in action recognition is the huge variations present in the data caused by the highly articulated nature of the human body. Human kinesics, being highly flexible in nature, results in high intra-subject and low inter-subject variabilities. This is further challenged by the variations introduced by the unconstrained nature of the environment where the video is captured. Since videos are composed of image frames, this introduces an additional dimension to the data, making it more difficult to define a model that properly focuses on the regions of interest that better discriminate particular action classes.

In order to mitigate these problems, one approach could be the design of a large scale dataset with fine-grained annotations covering the space of spatio-temporal variabilities defined by the problem domain, which would be infeasible in practice. The other approach is to design methods that can extract information from relevant spatial regions in the frames and encode the temporal variations occurring to these regions so that the video under consideration can be classified correctly.

1.1 Objective

This thesis work focuses on the problem of video representation learning, in the context of action recognition. One of the reasons why deep learning based action recognition methods under perform in comparison to their image recognition counterparts is due to the lack of large scale datasets with fine-grained annotations. However, developing large scale datasets with frame level annotations require a large amount of time and human effort deeming them practically implausible. The primary objective of this PhD work is to develop end-to-end trainable deep neural architectures for action recognition using video-level supervision alone. In order to achieve this, I approach the problem of video representation learning from two

fronts,

1. Encoding of temporal variations across the video frames
2. Encoding of relevant spatial regions in the video frames

As mentioned above, deep neural architectures such as ResNet [1], Inception [5], DenseNet [6] etc., are very competent in image classification. However, this competing image recognition performance has never been translated to video action recognition domain. Existing deep CNNs are capable of learning and generating effective feature representations of images. Videos are sequences of images with an inherent temporal dependency that defines the actual pattern to be recognized. Thus, existing CNNs can be used to generate feature representations of individual frames. In order to generate feature representation of a video, a technique has to be devised that can successfully encode the temporal changes occurring across the frame level features. The first goal of this thesis work is to explore and identify a suitable method that can encode the temporal variations occurring in a video.

As mentioned in the previous section, another challenge associated with action recognition problem is the variations in the environment where the videos are captured. In order to mitigate this issue, one approach is to develop a method that can identify and generate feature representation only from the relevant regions in the video frames. Neurophysiological studies have shown that human brain processes visual information selectively [7, 8]. This enables the brain to process information from relevant regions, such as the objects, and to suppress less relevant environmental information. The second goal of this PhD study is to develop techniques that can perform selective encoding of video frame features, mimicking the attention mechanisms present in the human visual cortex.

Since videos are sequences of images with temporal coherency, the spa-

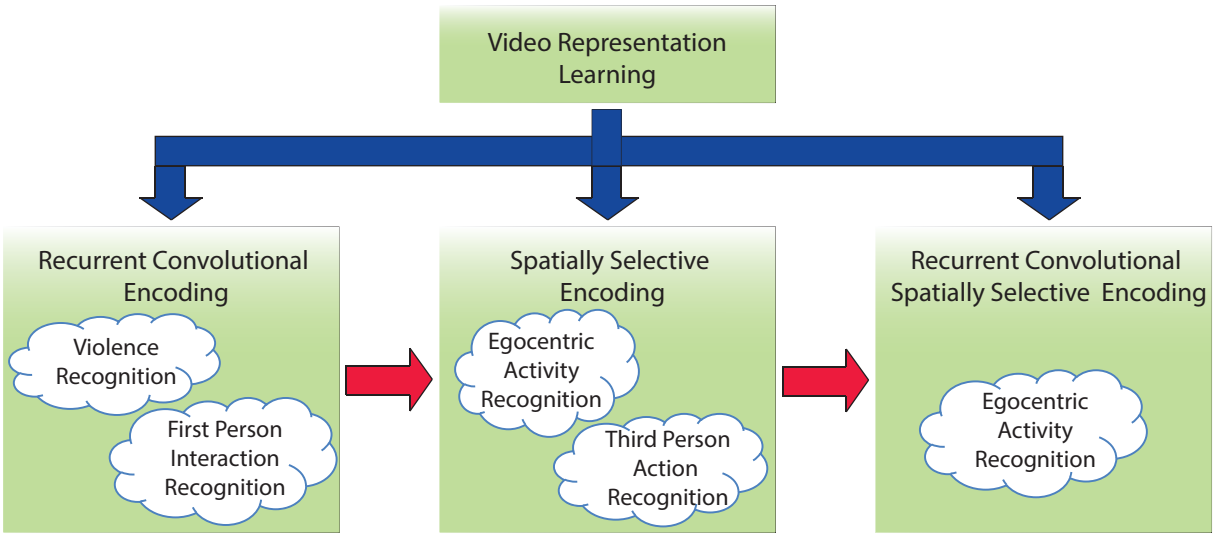


Figure 1.1: Overview of the thesis. The thesis explores the importance of temporal and spatial encoding for generating effective video representation.

tial attention maps generated from each frame should also reflect the same. So the third goal of this PhD study is to devise a technique that can effectively integrate the first and second goals such that the video representation generated takes into account the evolution of spatio-temporal features while maintaining a smooth tracking of the attention regions.

In order to evaluate the impact of each of the contributions for achieving the objectives, the corresponding proposed methods are tested on a variety of video classification tasks. This includes violence recognition, first person interaction recognition, egocentric activity recognition and third person action recognition.

1.2 Overview of the Thesis

Fig. 1.1 illustrates the overview of this thesis. As mentioned in Sec. 1.1, two approaches, effective temporal encoding and spatially selective encoding, are considered for addressing the problem of video representation learning.

Second chapter explores the first approach and perform a detailed anal-

ysis on Convolutional Neural Network (CNN)-Recurrent Neural Network (RNN) architectures for video representation generation that can be specialized to any supervised video classification task. This is followed by the development of two end-to-end learnable neural network architectures leveraging Convolutional Long Short-Term Memory (ConvLSTM) for encoding the temporal variations occurring in the video frames over time. The proposed architectures are evaluated on two tasks, violence recognition and first person interaction recognition, and their performance is compared against state-of-the-art techniques. These two tasks require strong temporal reasoning in order to be correctly classified and a detailed performance evaluation should reveal the benefit of CNN-RNN architectures in developing effective video representation. This chapter is based on the following publications [Pub1, Pub2].

An approach for performing spatial attention, based on the idea of Class Activation Maps (CAMs), to selectively encode features from relevant regions present in the video frame is presented in the third chapter. Performance of the proposed architecture is evaluated on two tasks, egocentric activity recognition and third-person action recognition. These two tasks are comparatively difficult than violence recognition and first person interaction recognition because of the presence of cluttered environments. As a result, the video encoding technique should be capable of aggregating features from relevant spatial regions. Detailed analysis of the proposed method reveals that selectively encoding features from relevant regions improves the recognition performance on the given task by obtaining state-of-the-art results. The findings of this chapter is published in [Pub3, Pub4, Pub6].

Chapter 4 first presents an in-depth analysis of Long Short-Term Memory (LSTM), a popular RNN module. Then a new recurrent neural unit, called Long Short-Term Attention (LSTA), is proposed that can perform

coupled temporal and spatially selective encoding while smoothly tracking the regions that are attended to in the previous frames. The effectiveness of the proposed module is evaluated on standard egocentric activity recognition benchmarks and obtain state-of-the-art results. This chapter is based on the results published in the paper [**?**].

Finally, the fifth chapter draws conclusions and discusses future research directions.

1.3 Contributions

This thesis makes the following contributions towards video representation learning:

- Presents an explorative study on Convolutional Neural Network (CNN)-Recurrent Neural Network (RNN) architectures for video representation generation and shows that convolutional recurrent architectures perform better than their Fully Connected (FC) counterparts, which were widely used for video representation generation;
- Proposes two architectures for recognizing violent videos and first-person interactions along with a detailed analysis;
- Proposes a spatial attention mechanism for selective encoding of features and performs a detailed study of the proposed approach for egocentric activity recognition and third person action recognition;
- Proposes a novel recurrent neural unit, called Long Short-Term Attention (LSTA), for generating video representation and perform a detailed analysis of the proposed module for the task of egocentric activity recognition.

Chapter 2

Recurrent Convolutional Encoding for Video Representation Learning

Videos are a series of successive images. A competent algorithm should take into account this succession or ordering of the images while generating a descriptor or a representation of the video. The generation of an order-preserving descriptor is vital in actions, such as `take` and `put`, `sit` and `standup`, `throw` and `catch`, etc., where shuffling of the sequence changes the action itself. Recurrent Neural Networks (RNNs) are a class of Neural Networks (NNs) capable of encoding sequence of vectors. The encoded vector will be different if the order of the sequence is changed. This is achieved by maintaining a hidden state or memory at each time step which is used in the processing of the subsequent vector in the input sequence. Thus, RNNs are a suitable canditate for encoding the feature descriptors, obtained from a CNN, of frames of a video. This chapter analyzes popular CNN-RNN architectures and evaluate their effectiveness on the task of violence recognition and first person interaction recognition.

The contributions of this chapter can be summarized as:

- Provides a brief overview of CNN-RNN architectures for video representation generation;

- Shows that a recurrent neural network with convolutional gates capable of encoding localized spatio-temporal changes generate a better representation, with less number of parameters and presents a novel CNN-RNN architecture by exploiting ConvLSTM for temporal encoding;
- Explores the possibility of using RGB frame difference as input for modeling the temporal variations occurring in a video sequence;
- Experimental validation of the performance of the proposed architecture on two applications, violence recognition and first person interaction recognition from videos.

The remaining of this chapter is organized in the following way: A brief overview of CNN-RNN architecture is presented in Sec. 2.1 followed by an introduction of ConvLSTM in Sec. 2.2. Secs. 2.3 and 2.4 develop two end-to-end trainable deep neural network architectures for recognizing violent videos and first person interactions from videos, respectively. Sec. 2.5 concludes this chapter.

2.1 CNN-RNN Architecture for Video Representation Learning

Fig. 2.1 illustrates a generic block diagram of CNN-RNN architecture. At each time step, the network will receive one frame from the video. The CNN backbone will generate a feature descriptor corresponding to the frame and is passed on to the RNN. This process is continued till all the frames of the video are applied to the network. Once all the frames are processed by the CNN and the RNN, the hidden state of the RNN is applied to a series of Fully Connected (FC) layers for classification. The advantages of such an architecture is threefold,

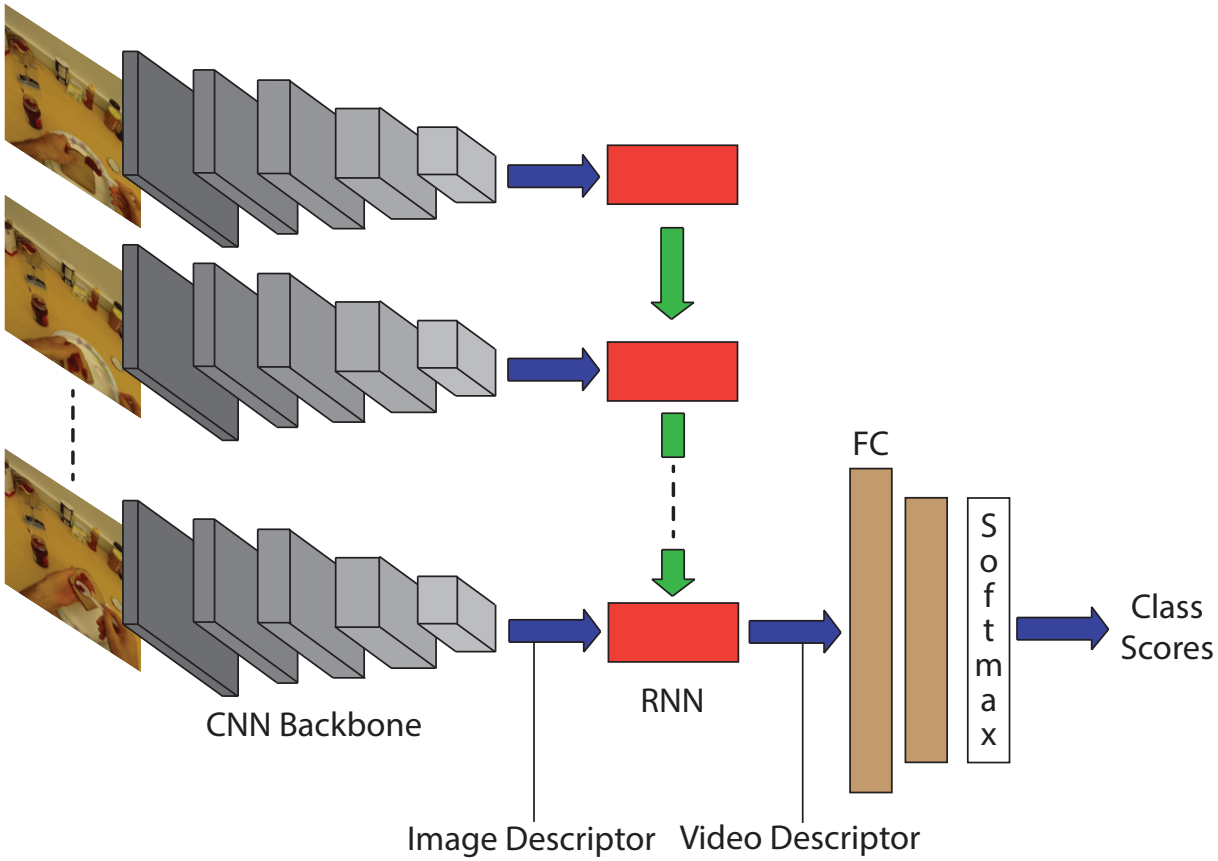


Figure 2.1: CNN-RNN architecture for video representation generation

1. It can perform encoding of sequences preserving their natural order;
2. It can be plugged into existing CNN architectures and trained end-to-end;
3. It can encode sequences of varying length.

However, the vanilla RNN suffers from the problem of vanishing and exploding gradients [9], thereby making them unsuitable for training sequences of longer length. To address these issues, two important variants of RNNs have been introduced, namely, Long Short-Term Memory (LSTM) [10] and Gated Recurrent Unit (GRU) [11]. They avoid the problem of vanishing gradient by adding a series of gated connections for controlling the hidden state of the RNN. The exploding gradient problem is

addressed by clipping the gradients. Both LSTM and GRU are found to be suitable for sequence encoding applications such as machine translation [12, 11], speech recognition [13, 14], handwriting recognition [15, 16], etc.

One of the pioneering work that proposed to use LSTM for sequential encoding of videos for action recognition is Long-Term Recurrent Convolutional Neural Network (LRCN) [17]. This model follows the same architecture explained above with an LSTM replacing the vanilla RNN. In this model, the output of the FC layer of a CNN are applied to an LSTM module for temporal encoding. Their model resulted in improved performances compared to a model using a single frame for classifying the video. However, this approach disregards a very important aspect of spatial features, the spatial structure of the video frame features. The FC layer operation results in the loss of spatial structure of the scene which is critical in understanding the contents of the image and how they evolve over time. In order to preserve the spatial structure of the convolutional features, one must encode the output of the final convolutional layer of the CNN rather than the output of the FC layer. This is prevented by the presence of FC gates of the LSTM.

2.2 Convolutional Long Short-Term Memory

Convolutional Long Short-Term Memory (ConvLSTM) is a variant of the LSTM architecture, in which the FC gates are replaced with convolutional gates. This model was proposed by Shi *et al.* [18] for addressing the problem of precipitation nowcasting prediction from radar images. The ConvLSTM is found to be functioning with improved performance compared to the FC LSTM for this task. The ConvLSTM model has been later used for predicting optical flow images [19] and anomaly detection [20]. The equations

governing the functioning of ConvLSTM are:

$$
\begin{aligned}
i_t &= \sigma(w_x^i * I_t + w_h^i * h_{t-1} + b^i) \\
f_t &= \sigma(w_x^f * I_t + w_h^f * h_{t-1} + b^f) \\
\tilde{c}_t &= tanh(w_x^{\tilde{c}} * I_t + w_h^{\tilde{c}} * h_{t-1} + b^{\tilde{c}}) \\
c_t &= \tilde{c}_t \odot i_t + c_{t-1} \odot f_t \\
o_t &= \sigma(w_x^o * I_t + w_h^o * h_{t-1} + b^o) \\
h_t &= o_t \odot tanh(c_t)
\end{aligned}
$$

where '$*$' and '$\odot$' represent the convolution operation and the Hadamard product, respectively, w_x^* are learnable weights and σ represents the Sigmoid non-linearity. In this all the gate activations and the hidden states are 3D tensors as opposed to the case with standard FC LSTM which are vectors. Because of the presence of convolutional gates, ConvLSTM can be inputted with convolutional feature tensors. This allows in the simultaneous encoding of both spatial and temporal features (spatio-temporal features) present in the video. The next two sections will evaluate the effectiveness of ConvLSTM compared to LSTM for two different video recognition tasks.

2.3 CNN-RNN for Detecting Violent Videos

Nowadays, the amount of public violence has increased dramatically. This can be a terror attack involving one or a number of persons wielding guns to a knife attack by a single person. This has resulted in the ubiquitous usage of surveillance cameras[1]. This has helped authorities in identifying violent attacks and take the necessary steps in order to minimize the disastrous effects. But almost all the systems nowadays require manual human

[1]There are around 100 surveillance cameras in the city of London (https://www.cityoflondon.police.uk/advice-and-support/Pages/Public-Space-Surveillance-Camera-System-.aspx)

inspection of these videos for identifying such scenarios, which is practically infeasible and inefficient. It is in this context that the proposed study becomes relevant. Having such a practical system that can automatically monitor surveillance videos and identify the violent behavior of humans will be of immense help and assistance to the law and order establishment. This work considers aggressive human behavior as violence rather than the presence of blood or fire.

2.3.1 Related Works

Several techniques have been proposed by researchers for addressing the problem of violence detection from videos. These include methods that use the visual content [21, 22], audio content [23, 24] or both [25, 26]. This section concentrates on methods that use the visual cues alone since it is more related to the proposed approach and moreover audio data is generally unavailable with surveillance videos. All the existing techniques can be divided into two classes depending on the underlying idea

1. Inter-frame changes: Frames containing violence undergo massive variations because of fast motion due to fights [27, 28, 29, 30]
2. Local motion in videos: The motion change patterns taking place in the video is analyzed [31, 32, 33, 21, 34, 35, 36, 37, 38, 22, 39, 40]

Vasconcelos and Lippman [27] used the tangent distance between adjacent frames for detecting the inter-frame variations. Clarin *et al.* improves this method in [28] by finding the regions with skin and blood and analyzing these regions for fast motion. Chen *et al.* [29] uses the motion vector encoded in the Moving Picture Experts Group (MPEG)-1 video stream for detecting frames with high motion content and then detects the presence of blood for classifying the video as violent. Deniz *et al.* [30] proposes to use the acceleration estimate computed from the power spectrum of adjacent frames as an indicator of fast motion between successive frames.

Motion trajectory information and the orientation of limbs of the persons present in the scene is proposed as a measure for detecting violence by Datta *et al.* [31]. Several other methods follow the techniques used in action recognition, i.e., to identify spatio-temporal interest points and extract features from these points. These include Harris corner detector [32], Space-Time Interest Points (STIP) [33], Motion Scale-Invariant Feature Transform (MoSIFT) [21, 35]. Hassner *et al.* [34] introduces a new feature descriptor called Violent Flows (ViF), which is the flow magnitude over time of the optical flow between adjacent frames, for detecting violent videos. This method is improved by Gao *et al.* [39] by incorporating the orientation of the violent flow features resulting in Oriented Violent Flows (OViF) features. Substantial derivative, a concept in fluid dynamics, is proposed by Mohammadi *et al.* [36] as a discriminative feature for detecting violent videos. Gracia *et al.* [38] proposes to use the blob features, obtained by subtracting adjacent frames, as the feature descriptor. The improved dense trajectory features commonly used in action recognition is used as a feature vector by Bilinski *et al.* in [22]. They also propose an improved Fisher encoding technique that can encode spatio-temporal position of features in a video. Zhang *et al.* [40] proposes to use a modified version of Motion Weber Local Descriptor (MoWLD), called Motion Improved Weber Local Descriptor (MoIWLD), followed by sparse representation as the feature descriptor.

The hand-crafted feature based techniques used methods such as bag of words, histogram, improved Fisher encoding, etc. for aggregating the features across the frames. Recently various models using LSTM RNNs [10] have been developed for addressing problems involving sequences such as machine translation [41], speech recognition [42], caption generation [43, 44] and video action recognition [17, 45]. The LSTM was introduced in 1997 to combat the effect of vanishing gradient problem which was plaguing

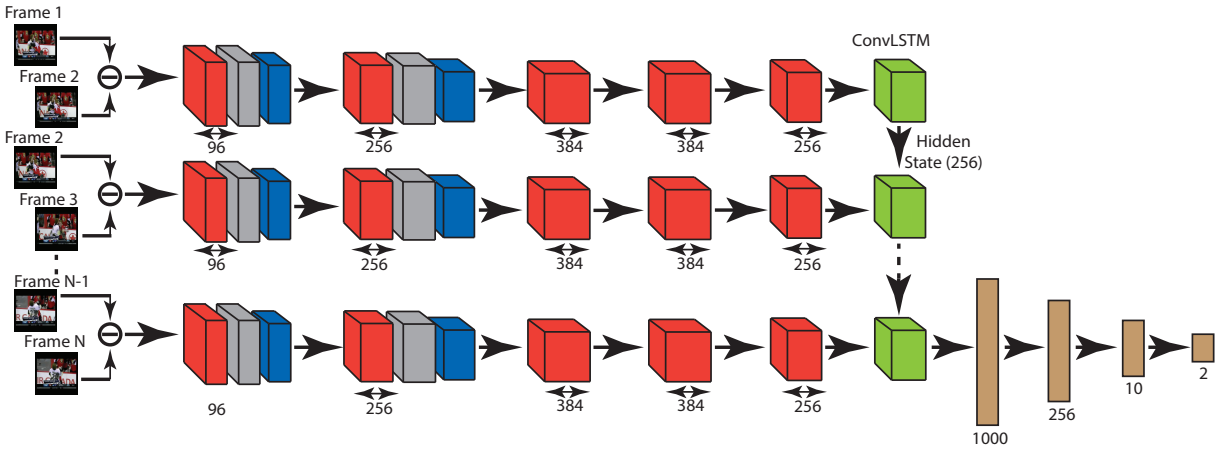


Figure 2.2: Block diagram of the proposed model. The model consists of alternating convolutional (red), normalization (grey) and pooling (blue) layers. The hidden state of the ConvLSTM (green) at the final time step is used for classification. The fully-connected layers are shown in brown colour.

the deep learning community. The LSTM incorporates a memory unit which contains information about the inputs the LSTM unit has seen and is regulated using a number of FC gates. The same idea of using LSTM for feature aggregation is proposed by Dong *et al.*in [46] for violence detection. The method consisted of extracting features using a CNN from raw pixels, optical flow images and acceleration flow maps followed by LSTM based encoding and a late fusion.

2.3.2 Proposed Method

The goal of the proposed study was to develop an end-to-end trainable deep neural network model for classifying videos into violent and non-violent ones. The block diagram of the proposed model is illustrated in figure 2.2. The network consists of a series of convolutional layers followed by max pooling operations for extracting discriminant features and ConvLSTM for encoding the frame level changes, that characterizes violent scenes, existing in the video.

For a system to identify a video as violent or non-violent, it should be capable of encoding localized spatial features and the manner in which they change with time. Hand-crafted features are capable of achieving this with the downside of having increased computational complexity. CNNs are capable of generating discriminant spatial features but existing methods use the features extracted from the FC layers for temporal encoding using LSTM. The output of the FC layers represents a global descriptor of the whole image. Thus the existing methods fail to encode the localized spatial changes. As a result, they resort to methods involving addition of more streams of data such as optical flow images [46] which results in increased computational complexity. It is in this context that the use of ConvLSTM becomes relevant as it is capable of encoding the convolutional features of the CNN. Also, the convolutional gates present in the ConvLSTM is trained to encode the temporal changes of local regions. In this way, the whole network is capable of encoding localized spatio-temporal features.

2.3.2.1 Network Architecture

Figure 2.2 illustrates the architecture of the network used for identifying violent videos. The convolutional layers are trained to extract hierarchical features from the video frames and are then aggregated using the ConvLSTM layer. The network functions as follows: The frames of the video under consideration are applied sequentially to the model. Once all the frames are applied, the hidden state of the ConvLSTM layer in this final time step contains the representation of the input video frames applied. This video representation, in the hidden state of the ConvLSTM, is then applied to a series of FC layers for classification.

The proposed model uses AlexNet model [47] pre-trained on the ImageNet dataset as the CNN backbone for extracting frame level features. Several studies have found out that networks trained on the ImageNet

dataset is capable of having better generalization and results in improved performance for tasks such as action recognition [48, 49]. In the ConvLSTM, 256 filters are used in all the gates with a filter size of 3×3 and stride 1. Thus the hidden state of the ConvLSTM consists of 256 feature maps. A Batch Normalization (BN) layer is added before the first fully-connected layer. Rectified Linear Unit (ReLU) non-linear activation is applied after each of the convolutional and FC layers.

In the network, instead of applying the input frames as such, the difference between adjacent frames are given as input. In this way, the network is forced to model the changes taking place in adjacent frames rather than the frames itself. This is inspired by the technique proposed by Simonyan and Zisserman in [48] to use optical flow images as input to a neural network for action recognition. The difference image can be considered as a crude and approximate version of optical flow images. So in the proposed method, the difference between adjacent video frames are applied as input to the network. As a result, the computational complexity involved in the optical flow image generation is avoided. The network is trained to minimize the binary cross entropy loss.

2.3.3 Experiments and Results

To evaluate the effectiveness of the proposed approach in classifying violent videos, three benchmark datasets are used and the classification accuracy is reported.

2.3.3.1 Implementation Details

The network is implemented using Torch library. From each video, N number of frames equally spaced in time are extracted and resized to a dimension of 256×256 for training. This is to avoid the redundant computations involved in processing all the frames, since adjacent frames contain

overlapping information. The number of frames selected is based on the average duration of the videos present in each dataset. The network is trained using RMSProp [50] algorithm with a learning rate of 10^{-4} and a batch size of 16. The model weights are initialized using Xavier algorithm [51]. Since the number of videos present in the datasets are limited, data augmentation techniques such as random cropping and horizontal flipping are used during training stage. During each training iteration, a portion of the frame of size 224×224 is cropped, from the four corners or from the center, and is randomly flipped before applying to the network. Note that the same augmentation technique is followed for all the frames present in a video. The network is run for 7500 iterations during the training stage. In the evaluation stage, the video frames are resized to 224×224 and are applied to the network for classifying them as violent or non-violent. All the training video frames in a dataset are normalized to make their mean zero and variance unity.

2.3.3.2 Datasets

The performance of the proposed method is evaluated on three standard public datasets namely, Hockey Fight Dataset [21], Movies Dataset [21] and Violent-Flows Crowd Violence Dataset [34]. They contain videos captured using mobile phones, CCTV cameras and high resolution video cameras.

Hockey Fight Dataset: Hockey fight dataset is created by collecting videos of ice hockey matches and contains 500 fighting and non-fighting videos. Almost all the videos in the dataset have a similar background and subjects (humans). 20 frames from each video are used as inputs to the network.

Movies Dataset: This dataset consists of fight sequences collected from movies. The non-fight sequences are collected from other publicly available action recognition datasets. The dataset is made up of 100 fight and 100

Input	**Classification Accuracy**
Frames (random initialization)	94.1±2.9%
Frames (ImageNet pre-trained)	96±0.35%
Frame Difference (random initialization)	95.5±0.5%
Frame Difference (ImageNet pre-trained)	97.1±0.55%

Table 2.1: Classification accuracy obtained with the hockey fight dataset for different models.

non-fight videos. As opposed to the hockey fight dataset, the videos of the movies dataset is substantially different in its content. 10 frames from each video are used as inputs to the network.

Violent-Flows Dataset: This is a crowd violence dataset as the number of people taking part in the violent events are very large. Most of the videos present in this dataset are collected from violent events taking place during football matches. There are 246 videos in this dataset. 20 frames from each video are used as inputs to the network.

2.3.3.3 Results and Discussions

Performance evaluation is done using 5-folds cross validation scheme, which is the technique followed in existing literatures. The model architecture selection was done by evaluating the performance of different models on the hockey fights dataset. The classification accuracies obtained for the two cases, RGB frames as input and RGB frame difference as input, is listed in table 2.1. From the table, it can also be seen that using a network that is pre-trained on the ImageNet dataset (we used BVLC AlexNet from Caffe model zoo) results in better performance compared to using a network that

Method	**Datasets**		
	Hockey	**Movies**	**Violent-Flows**
MoSIFT+HIK [21]	90.9%	89.5%	-
ViF [34]	82.9±0.14%	-	81.3±0.21%
MoSIFT+KDE+Sparse Coding [35]	94.3±1.68%	-	89.05±3.26%
Deniz *et al.* [30]	90.1±0%	98.0±0.22%	-
Gracia *et al.* [38]	82.4±0.4%	97.8±0.4%	-
Substantial Derivative [36]	-	96.89±0.21%	85.43±0.21%
Bilinski *et al.* [22]	93.4	99	**96.4**
MoIWLD [40]	96.8±1.04%	-	93.19±0.12%
ViF+OViF [39]	87.5±1.7%	-	88±2.45%
Three streams + LSTM [46]	93.9	-	-
Proposed	**97.1±0.55%**	**100±0%**	94.57±2.34%

Table 2.2: Comparison with state-of-the-art techniques for violence recognition.

is randomly initialized. Thus, it is decided to use frame difference as the input and to use a pre-trained network in the model. Table 2.2 gives the classification accuracy values obtained for the various datasets considered in this study and is compared against 10 state of the art techniques. From the table, it can be seen that the proposed method is able to better the results of the existing techniques in the case of hockey fights dataset and movies dataset.

As mentioned earlier, this study considers aggressive behavior as violent. The biggest problem of considering this definition occurs in the case of sports. For instance, in the hockey dataset, the fight videos consists of players colliding against each other and hitting one another. So one easy way to detect violent scenes is to check if one player moves closer to another. But the non-violent videos also consist of players hugging each other or doing high fives as part of a celebration. It is highly likely that these videos could be mistaken as violent. However, the proposed method is able to avoid this which suggests that it is capable of encoding motion

of localized regions (motion of limbs, reaction of involved persons, etc.). However, in the case of violent-flows dataset, the proposed method is not able to best the previous state of the art technique (it came second in terms of accuracy). Analyzing the dataset, it is found that in most of the violent videos, only a small part of the crowd is found to be involved in aggressive behavior while a large part remained as spectators. This forces the network to mark such videos as non-violent since majority of the people present in it is found to behave normally. Further studies are required for devising techniques to alleviate this problem involved with crowd videos. One technique that can be considered is to divide the frame into sub-regions and predict the output of the regions separately and mark the video as violent if any of the regions is outputted by the network as violent.

In order to compare the advantage of ConvLSTM over traditional LSTM, a different model that consists of LSTM is trained and tested on the hockey fights dataset. The new model consists of the AlexNet architecture followed by an LSTM RNN layer. The output of the last FC layer (fc7) of AlexNet is applied as input to an LSTM with 1000 hidden units. The rest of the architecture is similar to the one that uses ConvLSTM. The results obtained with this model and the number of trainable parameters associated with it are compared against the proposed model in table 2.3. The table clearly shows the advantages of using ConvLSTM over LSTM and the capability of ConvLSTM in generating useful video representation. It is also worth mentioning that the number of parameters that are required to be optimized, in the case of ConvLSTM, is very much less compared to LSTM (9.6M vs 77.5M). This helps the network to generalize better without overfitting in the case of limited data. The proposed model is capable of processing 31 frames per second on an NVIDIA K40 GPU.

Model	**Accuracy**	**No. of Parameters**
ConvLSTM	97.1±0.55%	9.6M(9619544)
LSTM	94.6±1.19%	77.5M(77520072)

Table 2.3: Comparison between ConvLSTM and LSTM models in terms of classification accuracy obtained in the hockey fights dataset and number of parameters

2.4 CNN-RNN for First Person Interaction Recognition from Videos

A new and vast array of research in the field of human activity recognition has been brought about with the recent technological advancements in wearable cameras. Majority of the human activity recognition techniques till now were concentrated on videos captured from a third person view. Developing techniques for the automatic analysis of ego-centric videos has immense application potential ranging from assistance during surgical procedure, law and order, assisting elderly citizens, video summarization, etc. But most of the existing research deal with activities carried out by the camera wearer, not interactions and reactions of others to the observer (person wearing the camera). The interaction recognition problem is different and difficult compared to the action recognition problem. This is because egocentric video captures a huge variety of objects, activities, and situations, and sometimes the person interacting with the observer can move out of the field of view as the person approaches the observer. Presence of ego-motion adds more complexity to the problem as it can interfere with the analysis of motion taking place in the scene. The applications of a method capable of automatically understanding interactions from egocentric view include surveillance, social interaction for robots, human-machine interaction, etc.

2.4.1 Related Works

An extensive amount of exploratory studies has been carried out in the area of first person activity or action recognition. They concentrate on the activity that the camera wearer is carrying out. These methods can be divided into two major classes: activities involving object manipulation such as meal preparation [52, 53, 54, 55, 56, 57] and activities such as running, walking, etc. [58, 59, 60, 61, 62]. The former relies on information about objects present in the scene for classifying the activity while the latter concentrates on the ego-motion and the salient motion in the scene.

More recent studies have focused on the analysis of interaction recognition in the egocentric vision context. One of the pioneering work in this area is carried out by Ryoo & Matthies in [63]. They extracted optical flow based features and sparse spatio-temporal features from the videos and a bag of words model as the video feature representation. Narayan *et al.* [64] use the TrajShape [65], Motion Boundary Histogram (MBH) [66] and Histogram of Optical Flow (HOF) [65] as the motion features followed by bag of words approach or Fisher vector encoding [67] for generating the feature descriptor. Wray *et al.* [68] propose graph-based semantic embedding for recognizing egocentric object interactions. Instead of focusing on the objects [69], Bambach *et al.* [70] investigate the use of strong region proposals and CNN classifier to locate and distinguish hands involved in an interaction.

Another line of research in this area was influenced by the development of Kinect device which is capable of capturing depth information from the scene together with the skeletal joints of the humans present in the scene. Methods exploiting this additional modality of data have been proposed by Gori *et al.*[71], Xia *et al.* [72] and Gori *et al.* [73]. Gori *et al.* [71] use the histogram of 3D joints, histogram of direction vectors and depth images

along with the visual features. Xia *et al.* [74] propose to use the spatio-temporal features computed from the RGB and depth images together with the skeletal joints information as the feature descriptor. Gori *et al.* [73] propose a feature descriptor called relation history image which extracts information from skeletal joints and depth images.

Several deep learning based approaches have been developed by researchers for action recognition from third person view. Simonyan & Zisserman [48] use raw frames and optical flow images as input to two CNNs for extracting the feature descriptor. Donahue *et al.* [17] and Srivastava *et al.* [45] use an architecture consisting of CNN followed by LSTM RNN for action recognition.

2.4.2 Proposed Method

Fig. 2.3 illustrates the architecture of the proposed approach. During each iteration two successive frames from the video will be applied to the model, i.e., frames 1 and 2 in the first time step, frames 2 and 3 in the second time step, etc. until all the video frames are inputted into the network. The hidden state of the RNN in the final time step is then used as the feature representation of the video and is fed to the classifier.

Simonyan & Zisserman [48] have previously proposed to use optical flow images along with RGB images, as the inputs to CNN, for performing action recognition which has resulted in improved performance. The optical flow images will provide information regarding the motion changes in the frames which will in turn help the network in learning discriminating features that can distinguish one action from another. Inspired from this, the performance of the network when the difference between adjacent frames are applied as input is also evaluated. The frame difference, obtained by subtracting the next frame from the current frame, can be considered as an approximate version of optical flow images. Sun *et al.* [75] and Wang

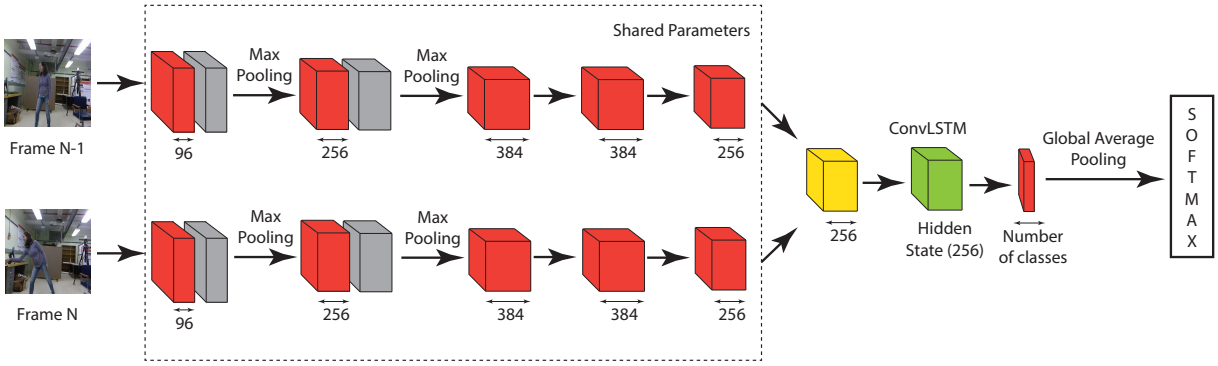


Figure 2.3: The architecture of the network. The convolutional layers are shown in red followed by normalization layer in gray. The 3D convolutional layer is shown in yellow and the convLSTM layer in green. Further experiments with a variant where, difference images obtained from pairs of successive frames instead of raw frames are also conducted.

et al. [76] have also explored the possibility of using difference images as inputs to a deep neural network for action recognition. By applying frame difference as the input, the network is forced to learn the changes taking place in the video, i.e., the motion changes which define the actions and interactions occurring inside the video.

2.4.2.1 Network Architecture

The architecture of the entire model used in the proposed approach is given in Fig. 2.3. The model can be divided into the following parts:

1. **Deep feature extractor**: A Siamese like architecture constituting of the AlexNet model [47] is used for extracting the frame level features. The fully-connected layers of the AlexNet model are removed. The two branches of the AlexNet model share their parameters and are pre-trained using the ImageNet dataset.

2. **Spatio-temporal feature aggregator**: This consists of a 3D convolution layer followed by the ConvLSTM explained in Sec. 2.2. The output of the two branches of the previous module are concatenated

together and is applied to the 3D convolution layer. The 3D convolutional layer uses a kernel of size $2 \times 3 \times 3$ and a stride of 1 in both the temporal and spatial dimensions. The output of the 3D convolution layer (256 feature planes) is then applied to the ConvLSTM layer for feature aggregation. The ConvLSTM layer uses 256 filters in all the gates with a filter size of 3×3 and stride 1. The hidden state, which contains the representation of the input video, consists of 256 feature planes.

3. **Classifier**: Once all the input frames are applied to the first two stages of the network, the video representation, which is the hidden state of the ConvLSTM, is applied to the classifier. The classifier consists of a spatial convolution layer followed by global average pooling. The convolution layer generates an output with feature planes corresponding to the number of classes. It uses 3×3 kernels with stride 1. Global average pooling is performed on each of the feature map generated by the convolutional layer and is applied to the softmax layer. This is inspired from the network in network (NIN) model [77] and the SqueezeNet model [78] proposed for image classification task.

The two branches of the AlexNet extract relevant frame level information consisting of the objects present in the frame. This will help the network in understanding a frame level context of the video. By using two branches of the AlexNet together with the 3D convolution layer, information in a short temporal duration is encoded. An advantage of following this approach evolves from the fact that in this way existing powerful CNN architectures that are pre-trained on large scale image datasets can be used, along with the capability to take into consideration the temporal context. The ConvLSTM layer maintains a memory of all the previous information it has seen which will result in the encoding of information in a longer

extend of time. This is very relevant in the case of egocentric videos. This is because, unlike in the case of third person views observing the scene from a distance, the objects or humans present in the video can move out of the field of view once they approach closer to the observer for physical interaction, which necessitates the requirement for a memory. Also, the convolutional gates present in the ConvLSTM helps the model to analyze the local spatial and temporal information together as compared to the FC LSTM. This has a significant importance in the analysis of videos because the network should be capable of understanding the spatial and temporal changes occurring in the video. By removing the fully-connected layers of the AlexNet and using the ConvLSTM, the network achieves this capability. More specifically, it maintains the spatial structure in the propagated memory tensor and the generated video descriptor. This is in contrast e.g. to LRCN [17] and the many CNN-LSTM architectures that fully convolve into a vector before temporal aggregation with standard LSTM, propagating a memory vector. This late spatio-temporal aggregation is distinguishing and more effective in representing spatial structure in videos. Furthermore, prior to classification it maintains the spatial structure till the very final layer of the architecture by applying spatial convolution to obtain class-specific activations that are directly associated to local receptive fields in the input. By following the classifier used in the proposed approach, a significant reduction in the number of parameters is achieved thereby minimizing the propensity to overfit. On top of that, the global average pooling by itself can be considered as a regularization [77], thereby avoiding the usage of other regularization techniques.

2.4.3 Experiments and Results

The performance, in terms of recognition accuracy, of the proposed approach is evaluated by testing it on four publicly available standard datasets.

2.4.3.1 Implementation Details

The implementation of the proposed method is done using Torch library. RMSProp [50] optimization algorithm is used for training the network with a fixed learning rate of 10^{-5} and a batch size of 12. The network is run for 10k iterations. The weights of the 3D convolution, ConvLSTM and the final spatial convolutional layers are initialized using Xavier method [51]. The whole network is trained together to minimize the cross-entropy loss.

Since it is a generally accepted consensus that deep neural networks tend to overfit in the absence of large amount of training data, several data augmentation techniques are used for minimizing the possibility of overfitting. In the proposed approach, all the video frames are first rescaled to a dimension of 256×340. Corner cropping and scale jittering techniques proposed by Wang *et al.* [76] are followed for data augmentation. In this, during each training iteration, a portion of the frame (four corners or the centre) is cropped and is applied as input to the network. The dimension of the cropped region is selected randomly from the following set {256, 224, 192, 168}. The cropped region is then rescaled to 224×224. The selected regions are then horizontally flipped randomly during each training iteration. It is to be noted that the same augmentation techniques are applied to all the frames coming from a video during a particular training iteration. All the video frames in the training set are normalized to make their mean zero and variance unity.

From all the videos, 20 frames equidistant in time are selected as the input to the network. During evaluation time, the mean of the training frames are subtracted from the testing frames and is divided by the variance of the former. Image crops of dimension 224×224 are extracted from all four corners and the center of each frame and are applied to the network. The horizontally flipped versions of the frames are also used during

evaluation. Thus, for each video in the test set, 10 different samples (5 crops and their horizontal flips) are generated during inference. The outputs of all the crops and their flipped versions are then averaged together to find the class of the input video.

2.4.3.2 Datasets

The following four datasets are used for validating the proposed first person interaction recognition approach.

JPL First-Person Interaction Dataset (JPLFPID) [63] : The dataset consists of videos of humans interacting with a humanoid model with a camera attached on top of its head. The dataset consists of 7 different interactions which include friendly (hugging, shaking hands, etc.), hostile (punching and throwing) and neutral (pointing fingers) behaviors. There are 84 videos in the dataset and the evaluation method is done following existing approaches. During each evaluation instance, half of the videos are chosen as training and the remaining half as testing. Mean of the recognition accuracy obtained after each evaluation is reported as final accuracy.

NUS First Person Interaction Dataset (NUSFPID) [64] : This dataset, captured by placing the camera on the head of a human, consists of both actions (opening door, operating cell phone, etc.) and interactions (shaking hands, passing objects, etc.) with other humans from a first person view point. The dataset contains 152 videos. Random train-test split method is used for evaluation and the mean of the accuracies obtained after all the runs is used for reporting.

UTKinect First Person Dataset-humanoid (UTKFPD-H) [72] : In this dataset, a kinect sensor is mounted on a humanoid robot and the video of various humans interacting with it is captured. This dataset also contains friendly (hugging, shaking hands, etc.), hostile (punching, throwing objects, etc.) and neutral behavior (standing up, running, etc.) behaviors.

There are 177 video samples from 9 different classes present in the dataset. The results reported are using the provided train-test split. For the experiments related to the proposed method, only the RGB videos are used and the depth information is not used.

UTKinect First Person Dataset-non-humanoid (UTKFPD-NH) [72] : This is similar to the above dataset with exception of the type of the robot used for capturing the video. A non-humanoid robot is used for capturing the videos in this dataset. The types of the interactions are also different from the previous dataset. There are 189 videos in the dataset and 9 different classes. Here also, the depth information available with the dataset is not used in the experiments.

All four datasets considered in this work is of significant difference from each other. For instance, the observer in each case is very much different so that the ego-motion that can occur to the video is different from one another. In the first dataset, the observer is stationary so that ego-motion can take place only if someone interacts with it physically. The NUSFPID and the UTKFPDH are more or less similar to each other but still the extend to which the interactions can affect their respective ego-motion is different, since the robot being more stable compared to a human. The type of interactions that are carried out in the datasets are also different. In this way, the experiments conducted are able to evaluate the performance of the proposed method in identifying interactions with different levels of ego-motion and different variety of interactions.

2.4.3.3 Results and Discussions

The results obtained for each of the dataset is reported and compared with the state of the art techniques in Tab. 2.4. As already mentioned, this is the first time a deep learning technique is proposed for recognizing first person interactions. Most of the existing deep learning based methods use gaze in-

Method	**Datasets**			
	JPLFPID	**NUSFPID**	**UTKinect First Person Dataset**	
			Humanoid	**Non-humanoid**
Ryoo & Matthies [63]	89.6	-	57.1	58.48
Ryoo [79]	87.1	-	-	-
Abebe *et al.*[80]	86	-	-	-
Ozkan *et al.*[81]	87.4	-	-	-
Narayan *et al.*[64]	**96.7**	61.8	61.9	57.6
Wang and Schmid [82]	-	58.9	-	-
HOF [63]	-	-	45.92	-
Laptev *et al.*[83]	-	-	48.46	50.83
LRCN [17] (raw frames)	59.5	68.9	72.6	71.4
LRCN [17] (frame difference)	89.0	69.1	63.1	67.8
Proposed Method (raw frames)	70.6	69.4	**79.6**	**78.4**
Proposed Method (frame difference)	91.0	**70.0**	66.7	69.1

Table 2.4: Comparison of the proposed method with existing techniques, on various datasets. Results are reported in terms of recognition accuracy in %. [64] on UTKinect are our reproduced results following the paper description. LRCN [17] results are produced from the authors' code following same training/testing protocol used of the proposed method. All other results are those available from the authors' papers.

formation or hand segmentations for identifying the objects being handled by the user which in turn is used for recognizing the action [56, 59]. Since first person interaction videos differ from such action recognition videos drastically, these methods are not considered for comparing the performance. It is evident from the table that the proposed method is suitable for recognizing interactions from first person videos. The proposed method outperforms the state of the art methods except in one dataset (JPL first person interaction dataset), where it came as the second best method. All

the state of the art methods listed in Tab. 2.4, except LRCN, use hand-crafted features as opposed to the proposed approach which is based on deep learning. As mentioned in the previous section, the videos in the JPL first person interaction dataset contain limited ego-motion as the camera is placed on a static object. Therefore, the ego-motion present in the videos is strongly correlated with the action that is occurring (for example, vertical motion when shaking hands, a sudden horizontal jerking motion in the punching action, etc.). Thus the ego-motion present in videos in the JPL dataset is part of the motion pattern that defines the action. In contrast to this, the rest of the datasets contain videos with strong ego-motion that may also arise occasionally and independently from the performed action. This may explain the results obtained on JPL dataset, where an improvement in performance is obtained when the frame difference is applied as the input to the network since the difference operation will force the network to model the motion changes between frames (including the ego-motion that depends on the action class). On the other hand, the performance of the network on the UTKinect dataset is the highest when the raw frames are applied as input. The videos in the UTKinect dataset contains strong ego-motion. The presence of ego-motion that is uncorrelated with the action being performed will act as a sort of noise when the frame difference is taken as input: the network should be able to distinguish between the random ego-motion and the motion caused by the action, thereby making it more difficult for the network to isolate the salient motion patterns occurring in the video. The network performs in a comparable way on the NUSFPID dataset when either types of inputs are applied. Even though the NUSFPID dataset contains ego-motion, some of the classes include actions performed by the observer such as using cell phone, typing on the keyboard, writing, etc. For discriminating these types of actions that contain limited amount of motion, the network needs to learn the objects

handled by the observer and hence the raw frames can be deemed as the most suited input modality. As opposed to this, interactions such as shaking hand, passing object, etc. contains motion and in this case, frame difference performs the best.

The proposed method is also compared with another deep learning approach proposed for action recognition, LRCN [17], in Tab. 2.4. As mentioned previously, LRCN uses an AlexNet model for extracting frame level features followed by an LSTM for temporal aggregation. The results show that the proposed method is more suitable for first person interaction recognition compared to the LRCN model. The primary reason is the usage of ConvLSTM in the proposed model as opposed to the FC LSTM used in the LRCN. In LRCN, the input of the LSTM is the output of the penultimate fully-connected layer (fc6). The local spatial information in the video frame is lost by using the FC layer, where as the proposed model preserves the spatial information by forwarding the convolutional features of the input. The convolutional gates of the ConvLSTM layer then performs information aggregation across both spatial and temporal domain. This is very important in the case of first person interaction recognition since it is required to memorize the changes occurring in both the spatial and temporal domains. Another feature of the proposed model that results in its improved performance is the utilization of the second AlexNet branch so that the model can process information from a broader temporal context. Since the parameters of both the branches are shared, there is no additional complexity added to the network and in addition to this, the same configuration can be used with any pre-trained CNN models. It should also be noted that, the proposed model achieves this improved performance with fewer number of parameters as compared to the LRCN model (21.8M ours vs 61.3M LRCN). This is achieved by eschewing from using FC layers (both in the ConvLSTM and the classification layers). This will also enable the

Method	**UTKFPD-H**	**UTKFPD-NH**
Xia *et al.* [74]	72.83	53.25
Oreifej & Liu [84]	52.54	45.55
Xia *et al.* [72]	85.6	83.7
Xia *et al.* [85]	-	70
Gori *et al.* [73]	-	85.94
Proposed Method	79.6	78.4

Table 2.5: Recognition accuracy obtained for the UTKinect first person dataset when the raw frames are applied as input to the proposed method. All other methods listed in the table uses either the depth image or skeletal joints information or both along with the RGB images for feature descriptor generation

proposed model to be trained on smaller datasets without getting overfit.

Tab. 2.5 compares the performance of the proposed method on the UTKinect dataset with state of the art methods that use other modalities of information such as depth image and skeletal joints. It is clearly evident from the table that the proposed method performs comparably to these methods without using any other additional data. The proposed method is able to perform competitively to these methods by using only the RGB video as input.

2.5 Conclusion

This chapter presented a brief overview of CNN-RNN architectures for video representation generation, specifically using LSTM and ConvLSTM RNN modules. It is found that, in order to effectively encode a video sequence, preserving the spatial structure of the frame level features during the temporal encoding stage is important. This is validated by evaluations made on two video recognition tasks, detection of violent videos and first person interaction recognition. Performance of the proposed model which uses ConvLSTM for temporal encoding consistently outperformed

the model that used LSTM. This confirms the effectiveness of ConvLSTM for encoding the variations of spatio-temporal features occurring over a video sequence.

Furthermore, it is also found that a CNN-RNN model can benefit from the use of frame difference as input as long as the camera motion is minimal. This was found to be much effective in the case of violent video recognition where the camera motion compared to the motion occurring in the video was less. Experiments conducted on the first person interaction datasets showed that presence of camera motion affects the performance of the network when frame difference is used as its input. This calls for further research into developing techniques for compensating the camera motion so that the advantages of frame difference can be leveraged for generating effective video representations.

Chapter 3

Spatially Selective Encoding for Video Representation Learning

In the previous chapter, we saw that ConvLSTM is effective in encoding frame level convolutional features from a deep CNN for generating video representations. This is achieved by propagating a memory tensor, inside the ConvLSTM, across each time step, thereby encoding the evolution of spatio-temporal features over time. The downside of this approach is that it considers all spatial regions to be equally important. This can result in the model learning the appearance features of irrelevant background information for classifying actions which can affect the model's generalization capability and introduction of bias to the model. In order to address this issue, I propose to use spatial attention in order to weight relevant regions present in the video frame which can assist in identifying the action taking place. The proposed attention mechanism is based on the idea of Class Activation Map (CAM), originally proposed for fine-grained image recognition and localization [86]. This chapter shows how to take advantage of the idea of CAM for generating attention maps and evaluates the effectiveness of this approach on popular egocentric activity recognition and third person action recognition benchmarks.

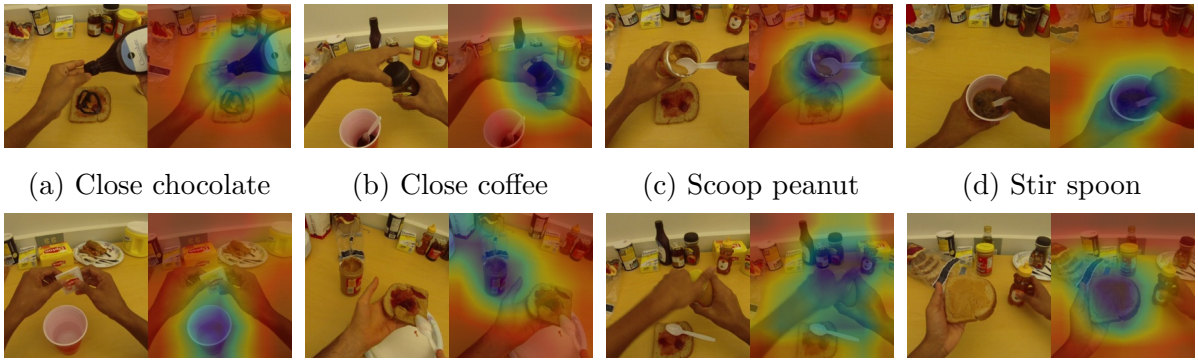
The main contributions of this chapter can be summarized as follows:

- Explores the relevance of spatially selective encoding via spatial attention for effective video representation generation;
- Develops two CNN-RNN architectures with spatial attention for egocentric activity and third person action recognition, respectively, that are end-to-end trainable using video-level supervision alone;
- Provides insights into the inner workings of the proposed models by visualizing the attention maps that are generated at the CNN-RNN interface of the architectures, and are found to be highly specialized and developed during training without hand segmentation and object location strong supervision and label semantics;
- Presents experimental validation on four egocentric activity recognition and two popular third person action recognition benchmark datasets along with an extensive ablation study.

The rest of this chapter is organized as follows. Sec. 3.1 presents a brief introduction of CAM. Sec. 3.2 presents an end-to-end trainable network with spatially selective encoding, called ego-rnn, for egocentric activity recognition along with a detailed performance evaluation. An end-to-end trainable model for third person action recognition, Top-down Attention Recurrent VLAD Encoder (TA-VLAD), is presented in Sec. 3.3. The chapter is concluded in Sec. 3.4.

3.1 Class Activation Mapping

Class activation Mapping is a technique which takes advantage of the average pooling layer present in modern deep CNNs such as ResNet [1], squeezeNet [78], denseNet [6], etc. for generating class specific saliency maps. This is performed by projecting back the weights of the output FC layer to the convolutional feature map in the preceding layer. Let $f_l(i)$ be



(a) Close chocolate (b) Close coffee (c) Scoop peanut (d) Stir spoon

(e) Open tea (f) Spread peanut (g) Open peanut (h) Take honey

Figure 3.1: Class Activation Maps obtained for frames from GTEA 61 dataset using ResNet-34 trained on ImageNet.

the activation of a unit l in the final convolutional layer at spatial location i and w_l^c be the weight corresponding to class c for unit l. Then the Class Activation Map (CAM) for class c, $M_c(i)$, can be represented as

$$M_c(i) = \sum_l w_l^c f_l(i) \tag{3.1}$$

Using the CAM, M_c, thus generated, one can identify the image regions that have been used by the CNN for identifying the class under consideration, which is class c. If the winning class is selected as c, i.e., the class category with the highest probability, then the CAM generated gives a saliency map of the image. One disadvantage of CAM is that it can be computed only on CNNs that contain a global average pooling layer preceding the FC classification layer. For this reason, a resNet-34 pre-trained on ImageNet dataset is chosen as the backbone CNN for frame level feature extraction and CAM generation.

Figure 3.1 shows some examples of CAMs generated using ResNet-34, pre-trained on ImageNet dataset, on some of the frames from GTEA 61 dataset [87]. The CAM generated is of the same spatial dimension as the features obtained from the final layer of the network (7×7 for ResNet-34). The CAM obtained is then resized to the input image dimensions for

visualization purpose. From the Figs. 3.1a, 3.1b, 3.1c, 3.1d, we can see that the network, which is pre-trained on ImageNet dataset, is capable of identifying the discriminant regions in the image which can be used for recognizing the activity class. However, in Figs. 3.1e, 3.1f, 3.1g, 3.1h, the regions obtained from the winning class of the CNN cannot be considered as a representative of the activity. Sec. 3.2.2.1, explains how to integrate and specialize CAM as spatial attention to identify the appropriate regions that can discriminate the activity under consideration from others.

3.2 Spatial Attention for Egocentric Activity Recognition

Automated analysis of videos for content understanding is one of the long standing research problems in computer vision. The majority of researchers working on video understanding problem concentrates on action recognition from distant or third person views while egocentric activity analysis has been investigated more recently. One of the major challenges in egocentric video analysis is the presence of ego-motion due to the frequent body movement of the camera wearer. This ego-motion may or may not be representative of the action performed by the observer and as a result, existing techniques proposed for general action recognition become less suitable for egocentric video analysis. Another reason hampering the development of this area is the lack of large scale datasets to enable the training of deep neural networks. However, with the ubiquitous availability and popularity of first-person cameras in recent times, providing more attention to this research area is of paramount importance.

This work takes on the problem of fine-grained recognition of egocentric activities, which is more challenging than egocentric action recognition. Action recognition involves identifying a generalized motion pattern

of hands such as `put`, `take`, `open`, `pour`, etc. whereas activity recognition concerns more fine-grained composite patterns such as `take bread`, `take water`, `put sugar in coffee`, `put bread in plate`, etc. For developing a system capable of recognizing activities, it is pertinent to identify both the hand motion patterns as well as the objects on to which a manipulation is being applied to. Majority of the state-of-the-art techniques use hand segmentation [60, 56, 59], gaze location [56] or object bounding boxes [60] for identifying the location of the relevant objects in the scene that can assist in identifying the activity. These approaches require complex pre-processing which includes human intervention for generating hand masks or gaze locations of the video frames. Even though there exist wearable devices which can estimate the gaze direction to dispose of the excessive cost of manual annotation, these may cause discomfort to the user or result in inaccuracies during distraction or short interruption of the activity or if the user is wearing glasses.

Considering the aforementioned problems that prevent from leveraging massive collections of natural activity videos, it is essential to develop techniques capable of identifying the relevant objects without being trained with full supervision. Towards this end, a CNN-RNN architecture that is trained in a weak supervision setting to predict the raw video-level activity-class label associated with the clip is presented. The CNN backbone of the proposed approach is pretrained for generic image recognition and augmented on top with an attention mechanism that uses CAMs for spatially selective feature extraction. The memory tensor of a ConvLSTM then tracks the discriminative frame-based features distilled from the video for activity classification. The design choices of the proposed approach are grounded to fine grained activity recognition because: (i) Frame-based activation maps are not bound to reflect image recognition classes, they develop their own representation classes implicitly while training the video-

level classification; (ii) ConvLSTM maintains the spatial structure of the input sequence all the way up to the final video descriptor used by the activity classification layer, thus facilitating the spatio-temporal encoding of objects and their locations into the descriptor as they develop into the activity over time.

3.2.1 Related Works

The majority of the methods developed for addressing egocentric activity recognition problem make use of semantic cues such as objects being handled, gaze direction, hand pose, ego-motion, etc. Fathi *et al.*[52] developed a hierarchical model based on object-hand configurations that makes use of information about object class, pose, hand optical flow, hand pose, etc. for egocentric activity recognition. In this, the model predicts actions based on the hand-object interactions which in turn is used to predict the activity. Fathi and Rehg [54] proposes to consider activity as a change of state of the objects and the materials present in the scene. A spatio-temporal pyramid approach in which the bin boundaries are sampled near the objects being handled by the user is proposed by McCandless and Grauman in [53]. Matsuo *et al.* [55] proposes a method to generate an attention map of the scene based on visual saliency and ego-motion information. The attention map is then used to classify the detected objects into salient and non-salient objects followed by a temporal pyramidal approach based feature extraction for activity recognition. Li *et al.* [56] proposes to encode egocentric cues along with dense trajectory features for activity recognition. They show that egocentric cues such as hand pose, hand motion, head movement, gaze direction, etc. can be used as efficient features for classifying egocentric activities.

Several deep learning based methods have been proposed for action recognition from third person videos in recent years. Majority of the ap-

proaches follow a two-stream network [48, 88, 89] consisting of an appearance stream encoding RGB images and a temporal stream encoding stacked optical flow. A number of methods adopting this two-stream architecture along with encoding of egocentric cues have been recently proposed. Singh *et al.* [59] adds an additional input stream, called ego-stream, to the two-stream network that takes in a stacked input of hand mask, saliency map and head motion for egocentric action recognition. A two-stream network consisting of an appearance stream trained for hand segmentation and object localization and a temporal stream for action recognition is proposed by Ma *et al.* [60]. The two individual networks are then jointly trained for recognizing fine-grained activities. Tang *et al.* [90] proposes to add an additional stream for encoding depth maps for improving activity recognition performance. Ryoo *et al.* [58] proposes to use a CNN for extracting frame level features which are then applied to a set of temporal filters that perform multiple pooling operations such as max pooling, sum pooling, histogram of gradients pooling, for generating effective feature descriptors.

Existing methods use a single RGB frame for extracting appearance information and neglect how the spatio-temporal changes evolve as the activity progresses. The proposed method uses ConvLSTM for encoding the spatio-temporal features present in the input video. As has been discovered by previous approaches, identifying the objects being handled by the user is useful in improving the performance of an activity recognition system. In the proposed approach, the idea of CAM proposed by Zhou *et al.* [86] is leveraged for identifying the location of the object that can be used for discriminating one activity class from another. The proposed approach builds on CAM as a spatial attention mechanism for weighting the features before encoding them.

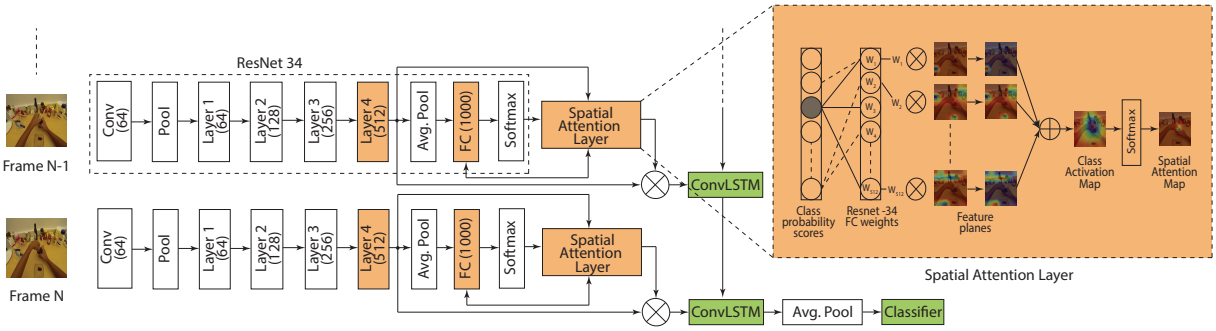


Figure 3.2: Block diagram of the proposed egocentric action recognition schema. ResNet-34 is used for frame-based feature extraction and spatial attention map generation, and ConvLSTM for temporal aggregation. Colored blocks indicate trained network layers, a two-stage strategy is followed: stage 1 green, stage 2 green+orange.

3.2.2 Proposed Method

This section explains the proposed method for egocentric activity recognition, consisting of deep feature encoding with spatial attention.

3.2.2.1 Spatial Attention Using Class Activation Maps

The block diagram of the proposed idea for encoding the RGB frames is shown in Fig. 3.2. In this, a ResNet-34 network pre-trained on ImageNet is used as the backbone architecture. For each RGB input frame, CAM is first computed using the winning class as given by equation 3.1. The CAM thus obtained is then converted to a probability map by applying the softmax operation along the spatial dimensions. This spatial attention map is then multiplied with the output of the final convolutional layer of ResNet-34, that is:

$$f_{SA}(i) = f(i) \odot \frac{e^{M_c(i)}}{\sum_{i'} e^{M_c(i')}} \tag{3.2}$$

where, $f(i)$ represents the output feature from the final convolutional layer of ResNet-34 at a spatial location i, $M_c(i)$ is the CAM obtained using the

winning class c, $f_{SA}(i)$ is the image feature after spatial attention is applied and $\odot$ represents the Hadamard product.

Once the image features with spatial attention (f_{SA}) is obtained, the next step is to perform temporal encoding of frame level features. A ConvLSTM module is used for performing this operation. As seen in the previous chapter, by using ConvLSTM for temporal encoding, it becomes possible to encode the changes in both temporal and spatial dimensions simultaneously. This is important since the network needs to track changes in both these dimensions if it is to learn how an activity evolves.

The spatial stream network works in the following way. During each time step, an input RGB frame will be applied to the network. The spatial attention will be computed and then image features with spatial attention is applied to the ConvLSTM module for temporal encoding. Once all the input frames are applied to the network, the memory of the ConvLSTM module, c_t is selected as the feature describing the entire video frames. Spatial average pooling operation is then applied on this c_t to obtain the video feature descriptor which is then fed to the classifier layer consisting of an FC layer for generating the class-category scores. A two-stage training strategy is followed in which only the classifier and the ConvLSTM layers are trained in the first stage followed by a second stage, where the convolutional layers of the final layer and the FC layer of the ResNet-34 network are also trained in addition to the ConvLSTM and classifier layers. This is represented using the colored blocks in Fig. 3.2. By training the convolutional layers and FC layer of the ResNet-34 network, the network's capability of localizing the relevant features for discriminating the activity will be greatly improved.

3.2.3 Experiments and Results

The proposed egocentric activity recognition method is evaluated on four datasets in terms of recognition accuracy.

3.2.3.1 Implementation Details

As mentioned previously, ResNet-34 pre-trained on ImageNet dataset is used as the base architecture for extracting frame level features and generating the object specific spatial attention map. A ConvLSTM module with 512 hidden units is used for temporal encoding. The network is trained in two stages as explained in Sec. 3.2.2.1. In the first stage, the network is trained for 200 epochs with an initial learning rate of 10^{-3} and the learning rate is decayed by a factor of 0.1 after 25, 75 and 150 epochs. Dropout at a rate of 0.7 is applied at the fully-connected classifier layer. In the second stage, the network is trained for 150 epochs with a learning rate of 10^{-4} and is decayed after 25 and 75 epochs by a factor of 0.1. ADAM optimization algorithm with a batch size of 32 is used during training. 25 frames from each video, uniformly sampled in time, is selected as the inputs to the network.

As is commonly done with deep learning based action recognition techniques [48, 88, 60, 59, 90, 89], the proposed approach also uses a temporal network that uses stacked optical flow images as input, in order to better encode the motion changes occuring in the input video. For this, a ResNet-34 pre-trained on ImageNet is trained with optical flow images. The cross-modality initialization approach proposed by Wang *et al.* in [88] is followed for initializing the input layer. In this approach, the weights of the first convolutional layer is initialized by computing the mean of the RGB pre-trained network's weights and replicating them by the number of channels of the target temporal network. The temporal network is trained

for 750 epochs with a batch size of 32 using Stochastic Gradient Descent (SGD) algorithm with a momentum of 0.9. The learning rate is fixed as 10^{-2} initially and is reduced by a factor of 0.5 after 150, 300 and 500 epochs. A stack of five optical flow images are applied as the input to the network and during evaluation, the average of the scores of 5 such stacks, uniformly sampled in time, is used to obtain the final classification scores. TV-L1 algorithm [91] is used for optical flow computation.

Once the spatial and temporal networks are trained, two approaches are followed for the fusion of these two. In the first approach, as proposed in [48], the scores obtained from each of the two networks is averaged to obtain the final classification score. In the second approach, the outputs of the two networks are concatenated and a new FC layer is added on top to get the class-category scores. Fine-tuning is then performed on all the layers down to the Layer4 of both networks for 250 epochs with a learning rate of 10^{-2}, which is reduced by a factor of 0.1 after each epoch. SGD algorithm is used as the optimization algorithm.

Since the amount of training samples are limited and to avoid overfitting, corner cropping, scale jittering and random horizontal flipping approaches as proposed in [88] are used for augmenting the data during training stages. During evaluation, the center crop of the frames are used to get the classification scores.

3.2.3.2 Datasets

As mentioned above, four standard benchmark datasets, GTEA 61 [87], GTEA 71 [87], GTEA Gaze+ [92] and EGTEA Gaze+ [93], are used to evaluate the performance of the proposed method. All the datasets are created in a controlled environment and consists of several activities involved in meal preparation.

GTEA 61 Dataset: This dataset consists of around 400 videos consisting

Configuration	Accuracy (%)
LRCN	34.48
ConvLSTM	51.72
ConvLSTM-attention	63.79
temporal-optical flow	44.83
temporal-warp flow	48.28
two-stream (average)	67.24
two-stream (joint train)	77.59

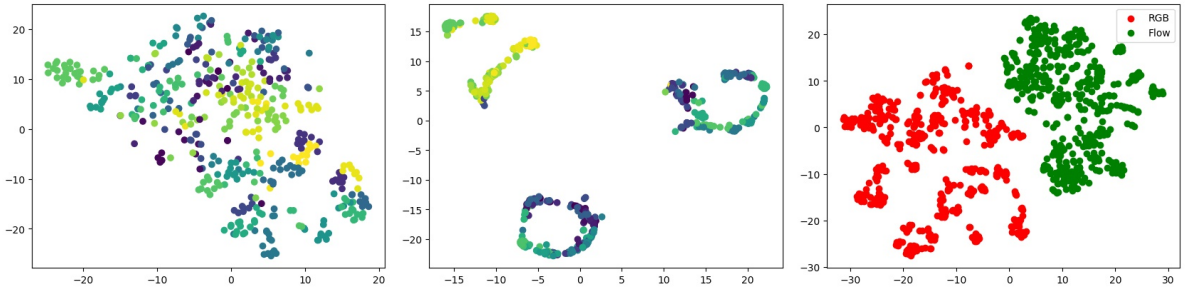
Table 3.1: Comparison of different configurations on the fixed split of GTEA 61 dataset.

of 61 classes. The videos are collected from 4 different subjects. Evaluation results are provided on both the fixed split (split 2) as well as leave-one-subject-out cross-validation setting.

GTEA 71 Dataset: This dataset is collected from the same videos used in GTEA 61 dataset and comprises of around 500 videos. The number of classes is 71. Evaluation is done based on the leave-one-subject-out cross-validation setting.

GTEA Gaze+ Dataset: This dataset is constructed from videos performed by up to 6 subjects. The dataset contains a total of approximately 1900 video segments with 44 activity classes. Leave-one-subject-out cross-validation setting is used to report the results.

EGTEA Gaze+ Dataset: EGTEA Gaze+ dataset is comparatively larger in size with 10k video samples from 106 activity classes. The videos are collected from 32 different subjects. This dataset subsumes the above three datasets. The results obtained on each of the three provided train-test splits as well as the average across the three splits are reported.



(a) ConvLSTM (b) LRCN (c) RGB vs Flow

Figure 3.3: t-SNE embedding of the features from (a) ConvLSTM model, (b) LRCN model and (c) spatial and temporal stream networks for videos from GTEA 61 dataset. In (a) and (b), features from videos of the same class have the same color.

3.2.3.3 Ablation Studies

The ablation study is performed on the various configurations using the fixed split of GTEA 61 dataset. The results obtained are listed in Tab. 3.1. For comparing the performance of FC LSTM with ConvLSTM, the LRCN approach proposed by Donahue *et al.* [17] is followed in which the output of the final convolutional layer of ResNet-34 is reshaped to obtain a feature descriptor describing the video frame. Feature descriptors thus generated from the frames of a video are then encoded using an LSTM with 512 hidden units. With the ConvLSTM module, as explained in Sec. 3.2.2.1, the proposed approach is able to keep the spatial dimensions and a memory tensor is propagated across time which enables in capturing the evolution of spatio-temporal changes with time. The advantages of using ConvLSTM can also be validated by comparing the recognition performance obtained with the respective models.The t-SNE embedding [94] of the features extracted from the output of both ConvLSTM and LRCN models are plotted in Fig. 3.3a and 3.3b. In Fig. 3.3a, the features are more spread out and those from the videos belonging to the same class are clustered together. From these observations, it is decided to use ConvLSTM as the tempo-

ral encoding module. Next, the performance gain obtained by adding the proposed spatial attention mechanism is analyzed and an improvement of 12% is observed by making the network attend to the relevant regions in the video frames.

Camera motion is found to be one of the problems degrading the performance of action recognition systems. In order to combat its effects, action recognition [88] and egocentric action recognition methods [56] propose to use stacked warp optical flow images in the temporal stream network. Warp optical flow is obtained by subtracting the camera motion from the optical flow. This is further validated by the experiments where an improvement of 4% is gained by compensating the camera motion in the optical flow images and as a result we choose to apply stacked warp optical flow images in the temporal stream of the network.

As explained in Sec. 3.2.3.1, two different approaches are followed for the fusion of spatial and temporal stream network outputs. In Tab. 3.1, two-stream (average) denotes the fusion approach where the class scores obtained from each of the two streams are averaged and two-stream (joint train) shows the result obtained by adding a new FC layer on top of the two individual networks followed by fine-tuning up to Layer 4. By following the joint training approach, a 10% performance boost is gained. This can be explained with Fig. 3.3c which plots the t-SNE embedding of the features obtained from the individual stream networks for videos from GTEA 61 dataset. From the Figure, it can be seen that the spatial stream and temporal stream features occupy in different regions of the feature space which indicates that they possess some complementary information that can assist in identifying one activity class from another. This resulted in the improved performance in the case of joint training since the network is allowed to learn to combine these complementary features in a way that each activity class becomes more discernible from one another.

Methods	Datasets			
	GTEA 61*	**GTEA 61**	**GTEA 71**	**GTEA Gaze+**
Li *et al.*[56]**	66.8	64	62.1	60.5
Ma *et al.*[60]**	75.08	73.02	73.24	**66.4**
Two stream [48]	57.64	51.58	49.65	58.77
TSN [88]	67.76	69.33	67.23	55.25
ego-rnn	**77.59**	**79**	**77**	60.13

Table 3.2: Comparison with state-of-the-art methods on popular egocentric datasets, we report recognition accuracy in %. (*: fixed split; **: trained with strong supervision)

3.2.3.4 Comparison with State-of-the-Art Techniques

Tabs. 3.2 and 3.3 compares the proposed method with state-of-the-art techniques. The first block in Tab. 3.2 shows methods specifically proposed for egocentric activity recognition, the second block shows methods proposed for third person action recognition. Regarding Tab. 3.3, EGTEA Gaze+ is a recently developed large scale egocentric video dataset comprising of 10325 activity instances from 106 classes. As the dataset is relatively new and no standard benchmarks are available, recognition performance is compared with state-of-the-art deep learning techniques for third person action recognition. Since this dataset could emerge as a future benchmark in egocentric activity recognition, the recognition accuracy obtained for each of the provided test-train splits is also listed in Tab. 3.3 for serving as a baseline in future research. All the compared methods in Tab. 3.2 and 3.3 use RGB images as well as optical flow images in order to classify the videos into the corresponding class categories.

The method proposed by Li *et al.* [56] uses hand segmentation for the first two datasets along with gaze information in the case of Gaze+ dataset for detecting the location of the objects being handled, while Ma *et al.* [60] trains a network for hand segmentation and object localization for ob-

Methods	Split 1	Split 2	Split 3	Average
Two Stream* [48]	43.78	41.47	40.28	41.84
I3D* [89]	54.19	51.45	49.41	51.68
TSN [88]	58.01	55.01	54.78	55.93
ego-rnn	**62.17**	**61.47**	**58.63**	**60.76**

Table 3.3: Recognition accuracy (in %) obtained for different train-test splits of EGTEA Gaze+ dataset compared against state-of-the-art techniques. (*: Result provided by the dataset developers)

taining explicit information about the objects and their location. In the proposed method, the prior knowledge inflated in the network is used to identify the location of the object that is relevant in identifying the activity class. From both Tabs. 3.2 and 3.3, we can see that the proposed method achieves a notable performance boost, except in the case of Gaze+ dataset, which anyway validates its efficacy in performing egocentric activity recognition. For Gaze+ dataset, the proposed method is resulting in lower performance than the methods proposed in [60] and [56]. This may be attributed to the nature of the dataset. In Gaze+ dataset, objects `spoon`, `fork` and `knife` are combined to form a meta object `spoonForkKnife` and objects `cup`, `plate` and `bowl` are combined as `cupPlateBowl`. Their corresponding activities are clubbed together as a single activity class. For example, activities such as `take knife`, `take fork` and `take spoon` are combined to form a single activity class titled `take spoonForkKnife`. In the proposed method, the network is trained to identify the relevant object's location from the video-level label alone, and combining different objects to form a new meta-object class weakens the correlation between activity class label and the visual information present in the frame. It therefore hampers fine-tuning of activation mapping during learning. This becomes more evident by observing the confusion matrix. Figs. 3.4 and 3.5

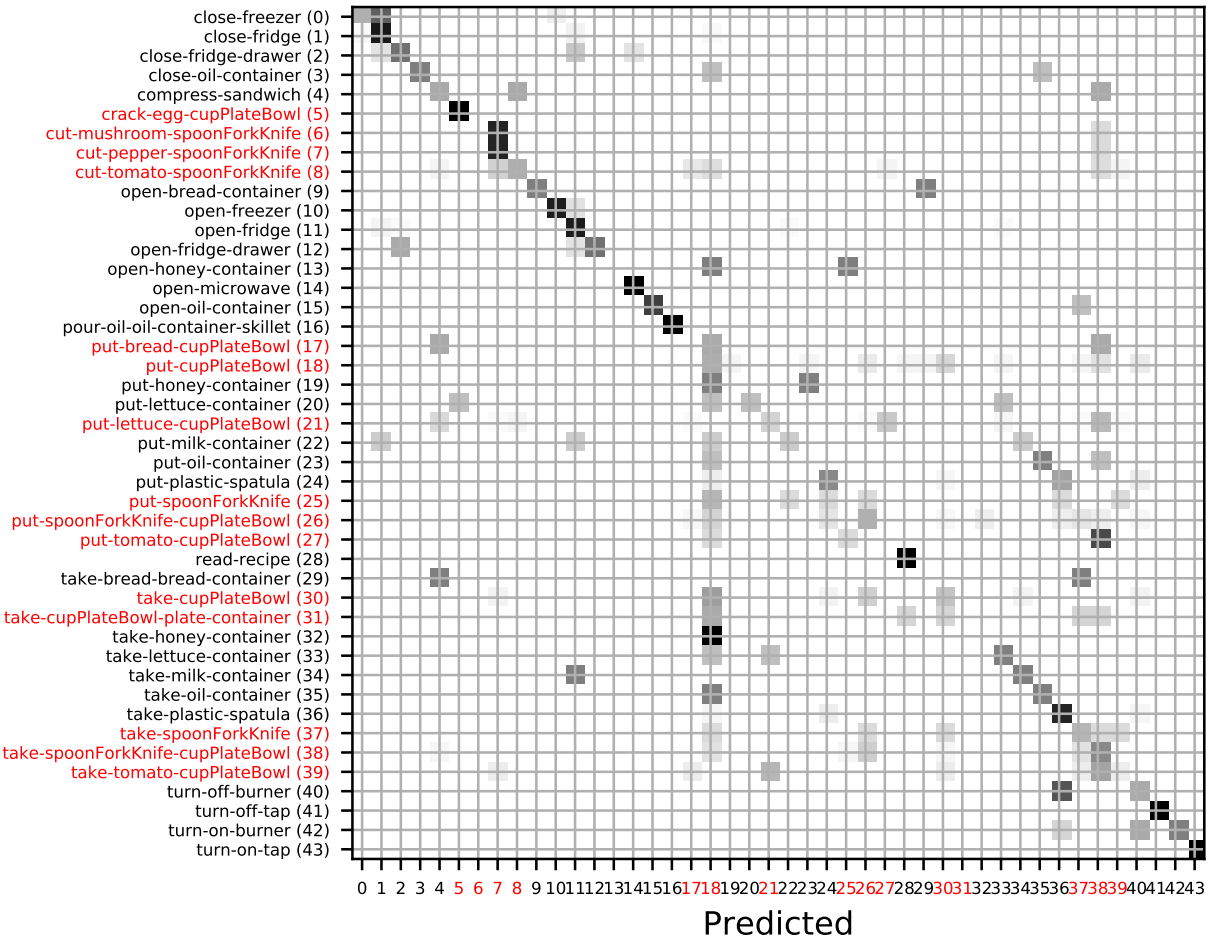


Figure 3.4: Confusion matrix of split P1 from GTEA Gaze+ dataset

shows the confusion matrices obtained when subjects P1 and P3 of the GTEA Gaze+ dataset are used as the test split. These two splits resulted in the least recognition accuracy out of all the splits. P1 gave an accuracy of 50.2% while P3 resulted in 48.84%. It can be seen that the accuracy of classes containing the meta-object labels (`spoonForkKnife` and `cupPlateBowl`) is low compared to the other classes. This further verifies the previously explained findings regarding the reason for the lower performance of the proposed method on GTEA Gaze+ dataset compared to the methods that use strong supervision for training.

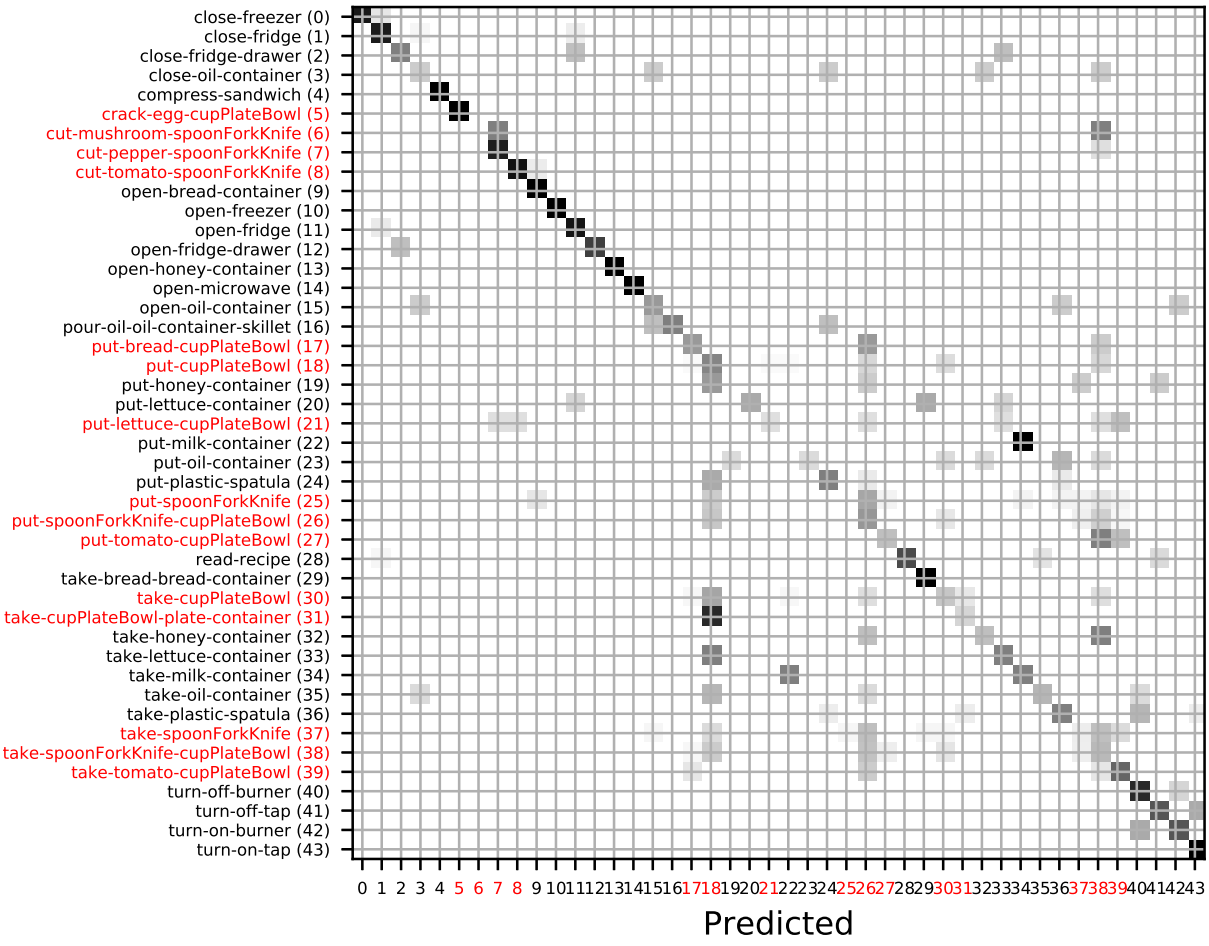
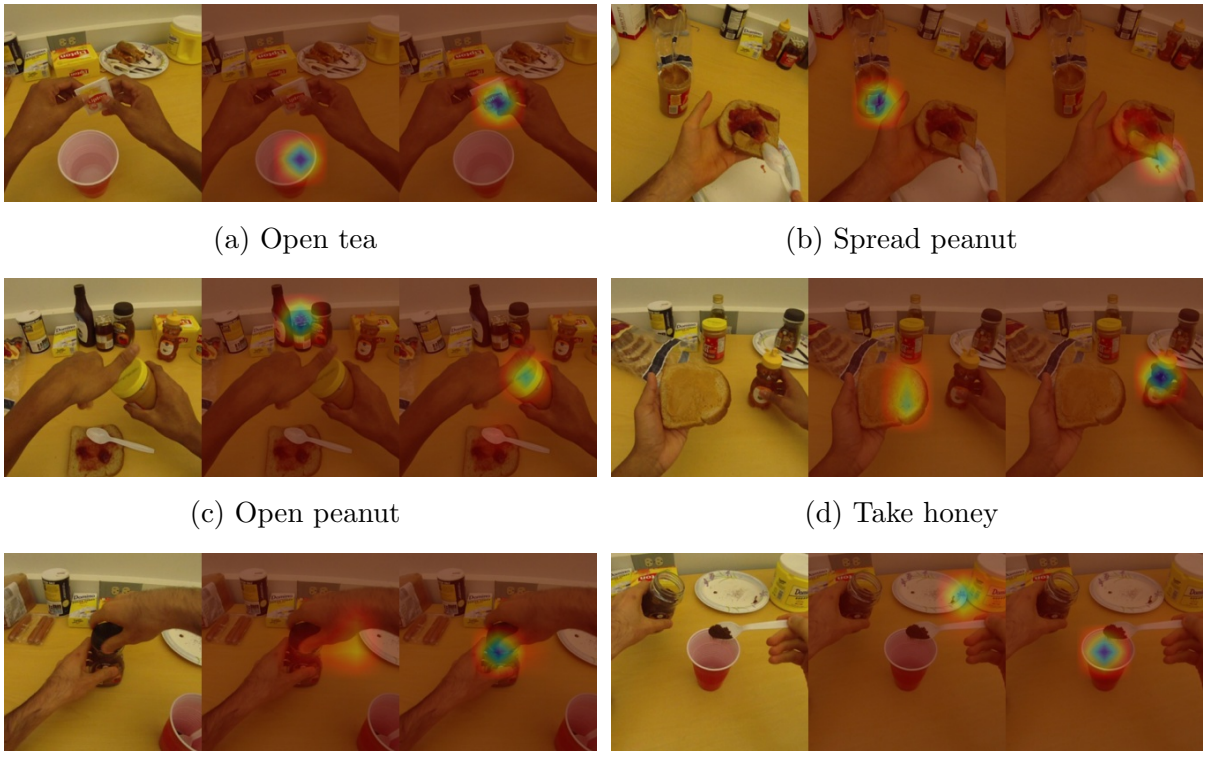


Figure 3.5: Confusion matrix of split P3 from GTEA Gaze+ dataset

In the case of methods proposed by Ma *et al.* [60] and Li *et al.* [56], the network is trained using explicit supervision in the form of object instance location, thereby resulting in improved performance compared to the proposed method. It should also be noted that a significant improvement is observed when comparing the proposed method with Two-Stream [48] on EGTEA Gaze+ dataset which subsumes Gaze+ dataset. In EGTEA Gaze+ dataset the above mentioned objects are not combined together to form a single class and the network more easily succeeds in identifying the regions in the video frames related to the activity.



(a) Open tea (b) Spread peanut

(c) Open peanut (d) Take honey

(e) Close Coffee (f) Pour Coffee

Figure 3.6: Spatial attention maps obtained for frames from GTEA 61 dataset. First image shows the original frame, second image shows the attention map generated with ResNet-34 trained on ImageNet and the last image shows the attention map obtained using the network trained for activity recognition (after stage 2 training).

In Fig. 3.1, we saw that a generic image recognition prior inflated in the network need not be representative of the activity class all the time. In order to compensate this problem, the FC layer of the ResNet-34 is trained together with the convolutional layers of the final layer of ResNet (orange blocks in Fig. 3.2) in stage 2. Fig. 3.6 shows the spatial attention map obtained after this second stage of training. From the Figure, we can see that the network learns to correctly identify the object in the images that is representative of the activity class under consideration.

Another point worth mentioning is that the state-of-the-art-techniques for action recognition from third person videos, listed in Tab. 3.2 and 3.3,

are performing sub-par compared to the methods proposed for egocentric videos. This shows that these methods cannot be considered as standardized techniques for action recognition from videos in general and underlines the importance of developing methods tailored for egocentric videos.

3.3 Spatial Attention for Third Person Action Recognition

This section presents a novel method for addressing the problem of third person action recognition. The previous section showed the importance of spatially selective encoding in generating effective video representation. However, that approach, ego-rnn, consisting of spatial attention for spatially selective encoding followed by ConvLSTM for temporal encoding, considered the feature tensor as a whole resulting in the encoding of motion changes in a global level. The actions carried out by a human need not necessarily be defined by the motion of his/her body as a whole but by the changes of certain body parts such as limbs or head. For *e.g.*, actions such as `kick` and `punch` are discriminated based on the movement of legs and hands, respectively. This calls for developing a technique that can encode the evolution of local features, over time, independently. In order to achieve this, this section proposes to leverage Vector of Locally Aggregated Descriptors (VLAD) encoding which transforms a feature tensor to a latent cluster representation by aggregating similar local features. By tracking the transition of these clusters via an RNN, it is possible to encode the local motion information across the frames. As mentioned above, certain actions are characterized by the movement of body parts of the actor. As a result, weighting these regions which discriminate one action category from another, leveraging the idea of spatial attention, is highly beneficial. Accordingly, this section presents Top-down Attention Recur-

rent VLAD Encoder (TA-VLAD), which uses (i) class specific activation maps obtained from a deep CNN pre-trained for image recognition as the spatial attention mechanism, a (ii) latent cluster representation of the feature space, and a (iii) GRU for temporal encoding in the cluster space. TA-VLAD can be trained end-to-end using video-level annotations, that is, the parameters of (i) and (iii) together with the compact representation of feature space (ii) are learned from videos paired with action class labels.

3.3.1 Related Works

3.3.1.1 Action Recognition

Compared to image recognition problem which requires encoding of spatial information alone, action recognition problem demands the encoding of spatio-temporal patterns present in multiple frames of a video. Several techniques have been proposed for extracting spatio-temporal information present in videos. Simonyan and Zisserman [48] propose to use stacked optical flow along with video frames for encoding both appearance and motion information present in the video. A huge performance improvement in terms of recognition accuracy was observed after incorporating the optical flow stream to the image based CNN, which confirms the fact that a simple CNN has limited capability in capturing spatio-temporal information. Wang *et al.* [95] propose to combine improved dense trajectory method with the two stream network of [48] to perform an effective pooling of the convolutional feature maps. The original two stream network [48] is further improved by adding residual connections from the motion stream to the appearance stream in [96]. Wang *et al.* [76] propose to use several segments of the video on a two stream network and predict the action class of each segment followed by a segment consensus function for predicting the action class of the video. This enables the network to model

long term temporal changes. A 3D convolutional network is proposed by Tran *et al.* [97] to encode spatio-temporal information from RGB images. Carreira and Zisserman [89] propose to combine this model with the two stream model to further improve the spatio-temporal information captured by the network. Several techniques that use RNNs such as LSTM [17, 98] and ConvLSTM [99, 100], have also been proposed for encoding long term temporal changes. Girdhar *et al.* [101] propose an end-to-end trainable CNN with a learnable spatio-temporal aggregation technique.

3.3.1.2 Attention for Action Recognition

A number of works have been recently proposed by researchers for video based action recognition incorporating attention mechanism [98, 102, 103, 104, 105, 106, 107]. Girdhar *et al.* [102] proposes an attentional pooling layer by extending the average pooling operation to weighted average pooling. Sudhakaran and Lanz [103] propose an object-centric attention model for egocentric activity recognition utilizing the class-specific activations from a CNN pre-trained for generic image recognition. The aforementioned approaches generate attention maps independently for each frame, hereby overlooking the temporal correlation between the frames of a video. This limitation is addressed in [98, 105, 104, 106, 107]. Sharma *et al.* [98] and Zhang *et al.* [105] propose to generate the attention weights from the hidden state of an RNN such as LSTM. Since the hidden state of the RNN encodes information about the previous frames, the attention maps are generated taking into consideration the information present in the previous frames of the video. Li *et al.* [104] propose to use a ConvLSTM to encode the spatio-temporal information and the attention map is generated from a single RGB frame and its corresponding optical flow. Du *et al.* [107] propose to use spatio-temporal attention in order to weight the relevant spatial regions in relevant frames. This is based on the idea that

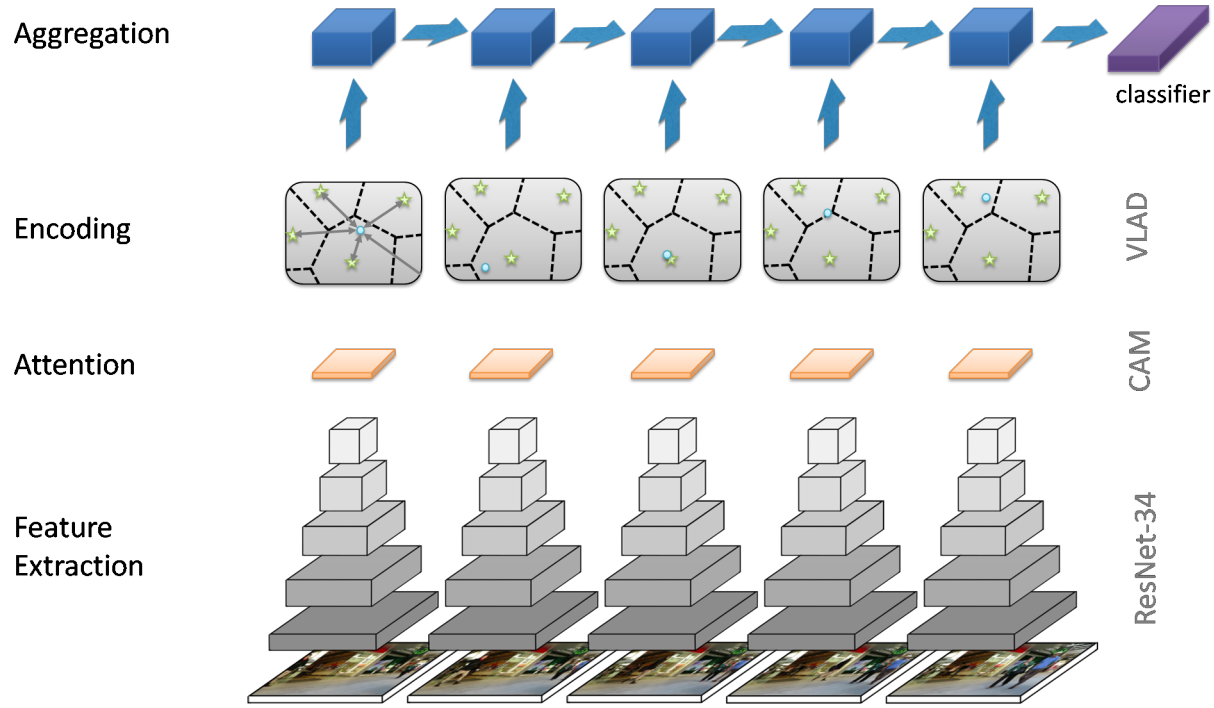


Figure 3.7: Block diagram of TA-VLAD architecture is given in the figure. A ResNet-34 is used for frame-based feature extraction and attention map generation, and temporally aggregated residual encoding to predict the action class from a frame sequence. All the layers of TA-VLAD are differentiable, enabling end-to-end training with video level supervision.

not all frames are equally important in identifying the action taking place in the video.

3.3.2 Proposed Method

This section details the proposed method named Top-down Attention Recurrent VLAD Encoder (TA-VLAD). TA-VLAD is built on the recently proposed ActionVLAD [101] method for action recognition. In [101], the authors develop an end-to-end trainable architecture that can perform VLAD aggregation of convolutional features extracted from a CNN. In order for the section to be self-contained, a brief explanation of VLAD encoding for action recognition is presented first followed by the details of

TA-VLAD.

3.3.2.1 VLAD Encoding

VLAD has been originally proposed for image retrieval application [108]. The method first generates a codebook of visual words c_k from local feature vectors extracted from a set of training images using k-means clustering. Given a new image with local feature vectors x_i of length P (i indexes location), the residual vectors $(x_i - c_k)$ are aggregated to form an image level descriptor V using membership values $a_k(x_i)$

$$V(j,k) = \sum_{i=1}^{N} a_k(x_i)(x_i(j) - c_k(j)) \tag{3.3}$$

where $j \in P$. The matrix V is then intra-normalized, reshaped into a vector and L2-normalized to obtain the final image descriptor. A key element here is the definition of membership value $a_k(x_i)$.

In the original paper [108] the feature vectors were hand-crafted and aggregated with hard assignment, that is, $a_k(x_i)$ will be 1 if x_i is closest to c_k and 0 otherwise.

The hard assignment was later replaced with a soft-assignment in [109] to form an end-to-end trainable CNN-VLAD network for place recognition. Their soft-assignment membership can be conveniently described by

$$a_k(x_i) = softmax_K(W_k^T x_i + B_k) \tag{3.4}$$

where $W_k = 2\alpha c_k$, $B_k = -\alpha||c_k||^2$ with $\alpha \gg 0$ being a hyper-parameter to control selectivity (for very large α it approximates to hard assignment). This operation is equivalent to applying a convolution layer with K filters of dimension 1×1 with weights and bias W and B, respectively. This enables the network to be end-to-end trainable with W and B as trainable parameters.

Later Girdhar *et al.* [101] extended this method to action recognition by performing summation of the frame level descriptors, that is

$$V(j,k) = \sum_{t=1}^{T} \sum_{i=1}^{N} a_k(x_{it})(x_{it}(j) - c_k(j)) \tag{3.5}$$

where T represents the total number of frames under consideration.

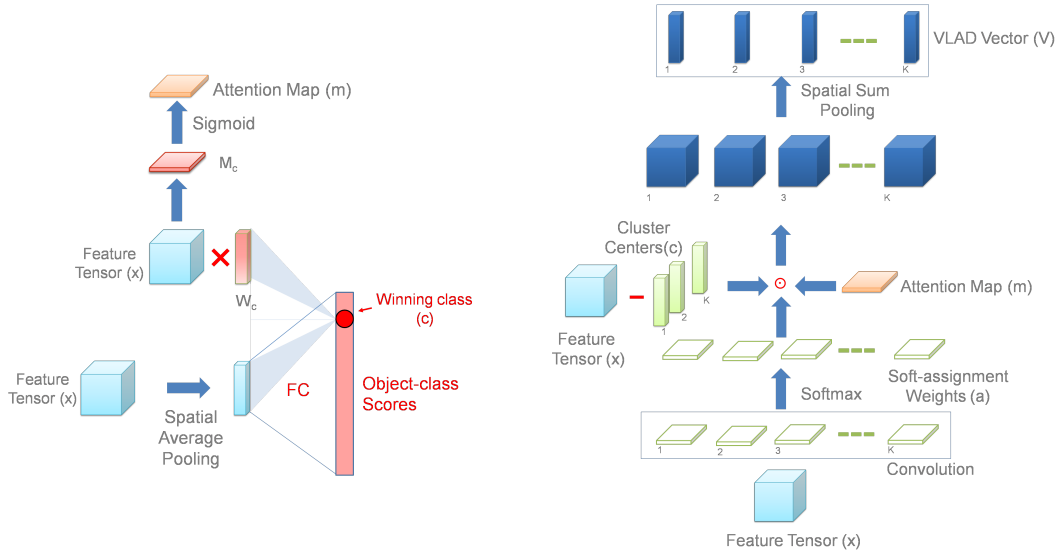
Kim and Kim [110] propose an extension to the original VLAD approach by weighting the descriptors depending on their importance. For determining the spatial image locations which contain discriminant information, a saliency map of the image is used. Their method resulted in performance improvements in image retrieval application. TA-VLAD is build upon this idea of weighted aggregation of VLAD descriptors for encoding visual features for action recognition. Instead of using a saliency detection algorithm, the proposed approach develops a deep neural network with built-in weighting mechanism whose parameters are trained and which adds only minimal computation overhead to training and inference stages of the whole pipeline. In the proposed approach, image descriptor is computed using Eq.(3.3) with membership values

$$a_k(x_{it}) = m_{it}\, softmax_k(W_k^T x_{it} + B_k) \tag{3.6}$$

where m_{it} is used to weight the residual $(x_{it} - c_k)$ at each spatial location i. This will be detailed in Sec. 3.3.2.2.

3.3.2.2 Network Architecture

This section describes in detail the proposed action recognition technique. A block diagram of the proposed approach is shown in Fig 3.8. In the proposed method, each frame from the input video, uniformly selected across time, are sequentially applied to the network. A deep CNN, ResNet-34 pretrained on ImageNet dataset, extracts the features from each frame. The



(a) Attention: Class activation mapping (b) Encoding: VLAD with attention soft-assignment

Figure 3.8: The proposed VLAD encoding and spatial attention map generation methods are shown in Figs. (a) and (b) respectively.

extracted feature tensor is then applied to the spatial attention map generation and VLAD encoding modules. The VLAD encoded feature vectors are then applied to a series of Gated Recurrent Units (GRUs) for temporal encoding. The memory of the GRU network obtained after the encoding of all the frames from the video is applied to a fully-connected layer for action classification. In the network, feature extraction from frames, top-down attention computation and VLAD encoding is carried out in a single forward pass. The purpose of top-down attention is to assign a higher weight to the regions present in the image that are useful for discriminating one action class from another while assigning a lower weight value to those regions that possess less discriminant information. Fig. 3.8a illustrates the attention map generation technique. The frame level feature tensor is first classified to ImageNet classes and the fully-connected weights corresponding to the winning class are used for generating the class activation map ($M_c(i)$), as described in Sec. 3.1. Top-down attention map is

obtained using the following equations

$$M_c(i) = \sum_l w_l^c f_l(i) \tag{3.7}$$

$$m_i = sigmoid(M_c(i)) \tag{3.8}$$

where c is the winning class, w_l^c is the weight in the final layer corresponding to class c, f_l is the activation at the final convolutional layer (input feature tensor). The *sigmoid* function is used to map the values of $M_c(i)$ in equation 3.7 to the range $[0, 1]$. The sigmoid function can be represented as

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3.9}$$

The proposed VLAD encoding method is illustrated in Fig. 3.8b. The feature tensor obtained from the CNN is applied to a convolutional layer followed by softmax operation to generate the soft-assignment weights (a_k) explained in Sec. 3.3.2.1. The weights and bias of convolutional layer is initialized with W_k and B_k, respectively, as given in Sec. 3.3.2.1. The soft-assignment weights are then multiplied with the attention map and the difference of the feature tensor and the cluster centers followed by summation operation across the spatial dimension to obtain a $K \times 512$ descriptor. Girdhar *et al.* [101] performed summation across the temporal dimension (summation of image descriptors obtained from all frames) for generating the video descriptor. This simple summation operation is not capable of encoding the temporal evolution of the frame level features as it does not take into account the temporal ordering of frames. Following the findings of Chapter 2, an RNN is chosen for this purpose. Chung *et al.* [111] have found that both LSTM [10] and GRU [11] perform comparably when evaluated on the tasks of polyphonic music modeling and speech signal modeling. With GRU having the added advantage of less parameters, it is decided to use GRU in order to effectively encode how the frame level features evolve as an action progresses. The commonly followed approach is

to flatten this $K \times 512$ tensor to a vector of dimension $1 \times K * 512$ and apply to a GRU layer. Following this approach will result in losing the structure of the local feature patterns encoded within the clusters since the gates in GRU layers combines information from multiple clusters during the encoding stage. The suggested alternative is to use K GRU layers, each encoding the residual information present in each of the K clusters. The disadvantage with this approach is the huge increase in the number of parameters of the resulting model, leading to increased computational complexity and overfitting. Taking into consideration the above mentioned problems, the proposed approach chooses to use K GRU modules with shared parameters, thereby having a separate memory state for each cluster, with less increase in the number of effective parameters. This enables the network to encode how the features associated with each of the clusters change with time. Since the clusters contain information about local features, the network is thus capable of tracking the local feature transformations occurring across the video frames. Once all the video frames are processed, the final memory state of all the GRU modules are concatenated to generate the temporally encoded descriptor. Then intra-normalization is applied as proposed in [112] followed by reshaping operation to obtain the vector and L2-normalization to get the final feature descriptor of the input video. This is followed by a single fully-connected layer for classification.

3.3.3 Experiments and Results

The proposed method is tested on two popular action recognition datasets, namely, HMDB51 [113] and UCF101 [114] and compared against state-of-the-art deep learning techniques.

3.3.3.1 Implementation Details

As mentioned in Sec. 3.3.2, ResNet-34 is used for feature extraction and attention map generation. The feature tensor obtained from the final convolutional layer of ResNet-34 (Layer4) as the input features. The proposed method is implemented using pyTorch framework. The method consists of the following steps:

1. *Pre-processing*, in which convolutional features from random frames from the videos present in the training set are extracted, followed by k-means clustering on the extracted features to obtain the cluster centers;

2. *Stage 1 training*, in which only the fully connected classifier layer and GRU network are trained while all the other parameters remain fixed;

3. *Stage 2 training*, in which all the convolutional layers in the final layer of ResNet-34 (layer 4) and fully-connected layer of ResNet-34, VLAD layers, GRU network and the classifier are trained.

Stage 1 training acts as an initialization of the classifier and the GRU network while stage 2 training optimizes the features extracted from the frames, the clusters, and the attention to specialize to the given action recognition task.

The number of clusters and the memory size of the GRU network are chosen as 64 and 256 respectively, following a detailed hyper-parameter search study. The network is trained for 30 epochs with a learning rate of 10^{-2} and 10^{-4} for stages 1 and 2 respectively. In stage 1, the learning rate is decayed by a factor of 0.5 after every 5 epochs while for stage 2 the learning rate is halved every 8 epochs. The network is trained using ADAM optimization algorithm with a batch size of 16. Dropout is applied at the

final FC layer at a rate of 0.5. The value of α, explained in Sec. 3.3.2.1, is chosen as 1000 following [101].

25 frames from each video clip that are uniformly sampled across time are used as inputs during training and evaluation. The corner cropping and scale jittering techniques proposed in [76] is used as data augmentation during training. The center crop of frames are used for determining the action class during evaluation.

3.3.3.2 Datasets

As mentioned above, two standard benchmark datasets are used for evaluating the effectiveness of the proposed method for third person action recognition.

HMDB51: HMDB51 dataset consists of videos divided into 51 action classes collected from movies and Youtube. The dataset is composed of 6849 video clips.

UCF101: UCF101 dataset consists of 13320 videos collected from Youtube and has 101 action categories. For both the datasets, three train/test splits are provided by the dataset developers. Following standard practice, the recognition performance on the datasets is reported as the average of the recognition accuracy obtained on the three splits.

Experiments are first performed on the first split of HMDB51 dataset to determine the suitable values for the number of clusters and the memory size of the GRU network. The results of the analysis is shown in Fig. 3.9a. From the graph, it can be seen that the best performance is obtained with a configuration having 64 clusters and 256 hidden units in the GRU. Thus, the number of clusters and the memory size of the GRU network are fixed as 64 and 256, respectively.

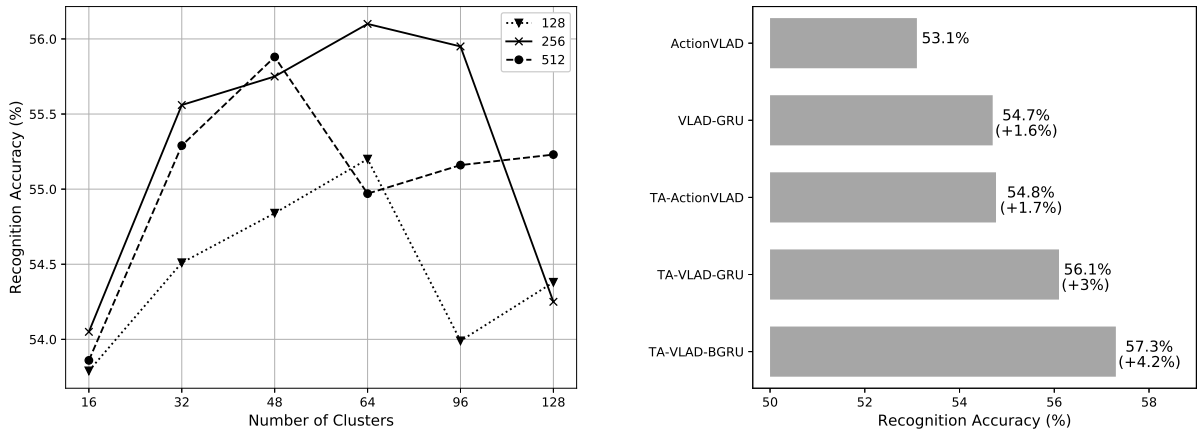
3.3.3.3 Model selection

Bidirectional RNNs have been proposed for applications involving sequences such as gesture recognition [115], speech recognition [13], named entity recognition [116] and machine translation [12]. This involves adding a second RNN layer where the input sequence is applied in the reverse direction. Inspired by the performance improvement obtained in the above mentioned works, evaluation of the proposed model is also done by adding a second GRU layer where the input is applied in the reverse direction. In the proposed method, the memory state of the two GRUs are concatenated and is then applied to the classification layer. An improvement of +1.2% is observed over the model with a single GRU encoding the sequence in the forward direction. However, the improvement observed was not this significant on the other splits of HMDB51 and on UCF101 dataset (Tab. 3.4). This can be explained by the nature of the data. In the case of applications involving language such as machine translation or speech recognition, the input at a particular time step may or may not depend on the input at a later time step. By encoding the sequence also in the reverse direction the network is able to look into the future and hence improve its performance. Also, in the proposed method, the output class is predicted from the final memory state of the GRU after encoding all the input frames as opposed to the above mentioned works, where an output is predicted at each time step. By using the final memory state, the network is able to encode all the information from the video sequence before making the prediction. On top of this, the additional GRU layer will increase the number of parameters of the network thereby increasing the memory footprint and the propensity to overfitting. The proposed model is also evaluated with bidirectional GRU having a memory size of 128, in order to have the same number of parameters as TA-VLAD-GRU and obtained a recognition accuracy of

56%.

For validating that training the layers of the backbone network used for CAM computation is useful, the ResNet-34 used for generating the attention map is decoupled from the rest of the trainable layers. With the attention generation branch not fine-tuned, a recognition accuracy of 55.3% is obtained on split 1 of HMDB51 dataset, as opposed to 56.1% with the fully trained TA-VLAD. This shows that joint training, as explained in Sec. 3.3.2, enables the network to improve the prior knowledge encoded within it, i.e., the relevance of objects and their locations in relation to the action performed. This can be interpreted from the example images shown in Fig. 3.10. From the figure, we can see that the network learns to attend to regions that contain discernible information about the action class. In addition, the network has adapted its attention for both actions containing objects (3.10b, 3.10c and 3.10d) as well as actions that are performed in the absence of objects (3.10a, 3.10e and 3.10g).

The performance of the network with a single GRU taking in the flattened VLAD vector of dimension $1 \times K * 512$ as input is evaluated and obtained a recognition accuracy of 47.6%. With this configuration the locally aggregated descriptors provided by VLAD encoding interact in the recurrent updating of a joint GRU memory state, hereby bypassing the learned latent structure (clustering) of the feature space. Indeed, a reduction of -8.5 percentage points in recognition accuracy is observed, validating the hypothesis that the flattening operation prevents the network from encoding the local spatial transformations effectively, as explained in Sec. 3.3.2.2. Analysis of the opposite model configuration is also performed, i.e. when learning to track a memory state for each feature space cluster independently. When the shared GRU in the proposed technique is replaced with K separate GRUs, the recognition accuracy was found to be 47.8%, that is a reduction of -8.3 percentage points. It is worth to note that the RNN



(a) Model selection

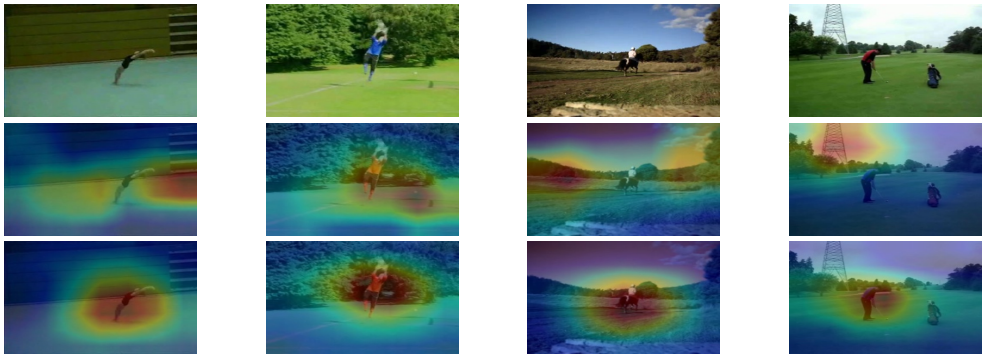
(b) Ablation study

Figure 3.9: The graphs show the results of the experiments conducted for (a) model selection and (b) ablation study. All the experiments are carried out on the first split of HMDB51 dataset.

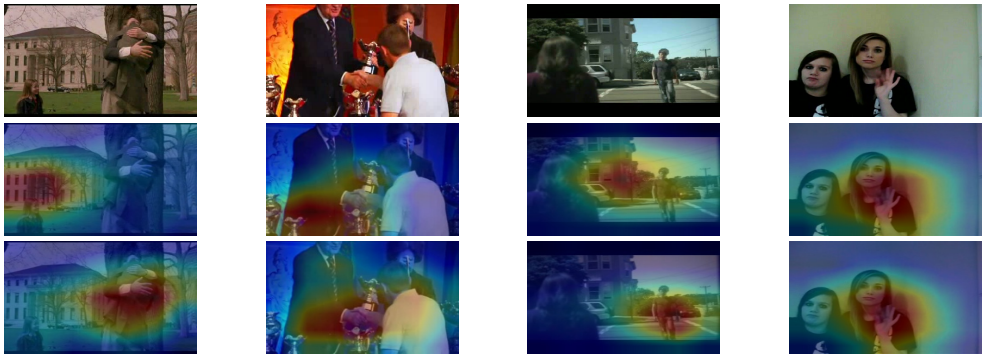
layer parameter count is increased by a factor of K ($K = 64$ in the proposed analysis), that may increase the network's propensity to overfitting. In TA-VLAD instead, all K aggregated descriptors contribute to the optimization of the single set of GRU parameters, and through parameter sharing the memory tracking observed during training for one cluster can be leveraged for memory tracking at any of the other clusters. This analysis shows that the proposed approach of using K GRU layers with shared parameters result in a most effective temporal aggregation of the frame level descriptors.

3.3.3.4 Ablation analysis

Next experiments are done to analyze the improvement obtained by the proposed method over the baseline, ActionVLAD [101], whose results are shown in Fig. 3.9b. In the experiments, the original ActionVLAD implementation is adapted by using Resnet-34 for fair comparison since TA-VLAD uses a more powerful and deep CNN, Resnet-34, for frame level



(a) Cartwheel (b) Catch (c) Ride horse (d) Golf



(e) Hug (f) Shake hands (g) Walk (h) Wave

Figure 3.10: Class Activation Maps for some frames in HMDB51 dataset with action classes containing objects (top) and without objects (bottom). Top row: original frames, second row: CAM obtained using ResNet-34 trained on ImageNet, bottom row: CAM obtained using the network trained for action recognition (after stage 2 training)

feature extraction compared to the VGG-16 used in [101]. Evaluation of the performance of the network after replacing temporal summation with the proposed temporal aggregation method consisting of GRU network (VLAD-GRU) is then performed. This resulted in an improvement of 1.6% (53.1 vs 54.7), which validates the hypothesis that preserving the temporal ordering of frame level feature descriptors is important for action recognition. In order to verify that spatially selective encoding of features using spatial attention is effective, the proposed attention mechanism is applied on ActionVLAD (TA-ActionVLAD) and evaluated the performance. An improvement of 1.7% (54.8%) is achieved over the Ac-

Method	**Backbone**	**Datasets**	
		HMDB51	**UCF101**
Two-Stream VGG [48]	VGG-M	40.5	73.0
Two-Stream ResNet [96]	ResNet-50	43.4	82.3
TDD [95]	VGG-M	50.0	82.8
I3D [89]	Inception V1	49.8	84.5
TSN [76]	Inception V2	51.0	85.1
ActionVLAD [101]	VGG-16	49.8	80.3
LSTM Soft Attention [98]	GoogleNet	41.3	84.9
Attentional pooling [102]	ResNet-101	50.8	-
VideoLSTM [104]	VGG-16	43.3	79.6
RSTAN [107]	Inception V2	53.4	80.2
TA-VLAD-GRU	ResNet-34	54.1	**85.7**
TA-VLAD-BGRU	ResNet-34	**54.9**	85.5

Table 3.4: Comparison of proposed approach against state-of-the-art methods on HMDB51 and UCF101 datasets (recognition accuracy in %).

tionVLAD baseline, thereby proving that not all spatial regions possess discriminant information for action recognition task. In the next step, the performance of the proposed approach, TA-VLAD that combines VLAD encoding with the proposed top-down attention mechanism and GRU sequence learning is evaluated, and an improvement of 3% (56.1%) is obtained over ActionVLAD.

3.3.3.5 Comparison to state-of-the-art

Table 3.4 compares the proposed approach, TA-VLAD, with state-of-the-art techniques. The results are the average of the recognition accuracy in % obtained over all three splits. For fair comparison, the performance of the compared methods using RGB frames only are reported and consider those methods that are based on deep learning and use ImageNet for pre-training. For each of the three splits in HMDB51, TA-VLAD obtained recognition

accuracies of 57.3%, 53.1% and 54.4% and on UCF101 dataset, 85.7%, 85.5% and 85.2%. The first block on the table shows methods which do not use attention and the second block shows those that use attention for selecting the relevant regions present in the input frames. From the table, it can be seen that the proposed approach, TA-VLAD performs better than the state-of-the-art deep learning methods for action recognition on RGB image sequences.

3.4 Conclusion

This chapter presented the importance of weighting spatial regions of feature tensors before temporal encoding along with a novel method for achieving this. Based on the proposed idea that leverages CAM, two end-to-end trainable deep neural architectures have been proposed for addressing egocentric activity recognition and third person action recognition problems. Detailed performance evaluations performed on standard benchmarks reveal the effectiveness of spatially selective encoding for video representation generation. The proposed approach takes advantage of the prior information encoded in a CNN pre-trained for image classification to generate object specific spatial attention. Detailed analysis show that the proposed spatial attention mechanism is able to correctly locate the objects/regions that are representative of the activity/action class under consideration. In both the proposed architectures, the networks were able to steer the attention away from ImageNet prior to discriminative locations that explain the relation of input frame regions to the video-level label. This confirms that the architectures have good enough gradient flow property to identify and locate the relevant spatial regions when trained using video-level supervision alone.

Chapter 4

Recurrent Convolutional Spatially Selective Encoding for Video Representation Learning

In the previous chapter we saw the importance of spatially selective encoding via attention mechanism for video representation generation. The attention mechanism proposed in the previous approach was successful in recognizing fine-grained egocentric activities and third person actions. The proposed method was also able to shift the attention from ImageNet prior to task-specific attention regions when trained on video-level supervision alone making the approach pragmatic for video classification since developing large scale datasets with frame-level annotations is practically infeasible. However, in that approach, the attention is computed individually for each frame which can lead to instability in activation map selection owing to the network selecting different winning classes in the adjacent frames of the same video. Consequently, the model may lose a proper smooth tracking of the attended regions resulting in the model attending to different locations in different frames from the same video thereby failing to encode features from relevant spatial regions that can discriminate one class category from another.

To address this limitation, in this chapter, a detailed study on the more general question of how a video CNN-RNN can learn to focus on the regions of interest to better discriminate the action classes is done. This is done by analyzing the shortcomings of LSTMs in this context and deriving Long Short-Term Attention (LSTA), a new recurrent neural unit that augments LSTM with built-in spatial attention and a revised output gating. The first enables LSTA to attend the feature regions of interest while the second constraints it to expose a distilled view of the internal memory, which represents the evolution of spatio-temporal features in the video. This second aspect is found to be effective in the recurrent unit since output gating does not only affect prediction overall but controls the recurrence, being responsible for a smooth and focused tracking of the latent memory state across the sequence.

The main contributions of this chapter can be summarized as follows:

- Presents Long Short-Term Attention (LSTA), a new recurrent unit that addresses shortcomings of LSTM when the discriminative information in the input sequence can be spatially localized;

- Deploys LSTA into a two stream architecture with cross-modal fusion, a novel control of the bias parameter of one modality by using the other;

- Reports an extensive ablation analysis of the model and evaluates it on egocentric activity recognition, providing state-of-the-art results in four public datasets.

The remaining of this chapter is organized as follows. A detailed analysis of LSTM is given in Sec. 4.1. The proposed RNN module, LSTA, is presented in Sec. 4.2. Sec. 4.3 analyzes the effectiveness of LSTA for egocentric activity recognition. The chapter is concluded in Sec. 4.4

4.1 Analysis of LSTM

LSTM is the widely adopted neuron design for processing and/or predicting sequences. A latent memory state $\mathbf{c_t}$ is tracked across a sequence with a forget-update mechanism

$$\mathbf{c_t} = f \odot \mathbf{c_{t-1}} + i \odot c \tag{4.1}$$

where (f, i) have a gating function on the previous state $\mathbf{c_{t-1}}$ and an innovation term c. (f, i, c) are parametric functions of input $\mathbf{x_t}$ and a gated non-linear view of previous memory state $\mathbf{o_{t-1}} \odot \eta(\mathbf{c_{t-1}})$

$$(i, f, \mathbf{o_t}, c) = (\sigma, \sigma, \sigma, \eta)(W \cdot [\mathbf{x_t}, \mathbf{o_{t-1}} \odot \eta(\mathbf{c_{t-1}})]) \tag{4.2}$$

The latter, referred to as hidden state $\mathbf{h_t} = \mathbf{o_t} \odot \eta(\mathbf{c_t})$, is often exposed to realize a sequence prediction. For sequence classification instead, the final memory state can be used as a fixed-length descriptor of the input sequence.

Two features of LSTM design explain its success. First, the memory update (Eq. 4.1) is flexibly controlled by (f, i): a state can, in a single iteration, be erased $(0, 0)$, reset $(0, 1)$, left unchanged $(1, 0)$, or progressively memorize new input. $(1, 1)$ resembles residual learning [1], a key design pattern in very deep networks - depth here translates to sequence length. Indeed, LSTMs have strong gradient flow and learn long-term dependencies [10]. Second, the gating functions (Eq. 4.2) are learnable neurons and their interaction in memory updating is transparent (Eq. 4.1). When applied to video classification, a few limitations are to be discussed:

1. Memory: Standard LSTMs use fully connected neuron gates and consequently, the memory state is unstructured. This may be desired *e.g.* for image captioning where one modality (vision) has to be translated into another (language). For video classification it might be

advantageous to preserve the spatial layout of images and their convolutional features by propagating a memory tensor instead. ConvLSTM [18] addresses this shortcoming through convolutional gates in the LSTM.

2. Attention: The discriminative information is often confined locally in the video frame. Thus, not all convolutional features are equally important for recognition. In LSTMs the filtering of irrelevant features (and memory) is deferred to the gating neurons, that is, to a linear transformation (or convolution) and a non-linearity. Attention neurons were introduced to suppress activations from irrelevant features ahead of gating. The proposed module augments LSTM with built-in attention that directly interacts with the memory tracking and is explained in Sec. 4.2.1.

3. Output gating: Output gating not only impacts sequence prediction but it critically affects memory tracking too, *cf.* Eq 4.2. The new module replaces the output gating neuron of LSTM with a high-capacity neuron whose design is inspired by that of attention. There is indeed a relation among them, which is made explicit in Sec. 4.2.2.

4. External bias control: The neurons in Eq. 4.2 have a bias term that is learnt from data during training, and it is fixed at prediction time in standard LSTM. A suitable approach is required for adapting the biases based on the input video for each prediction. State-of-the-art video recognition is realized with two-stream architectures, the proposed approach uses flow stream to control appearance biases in Sec. 4.3.2.3

4.2 Long Short-Term Attention

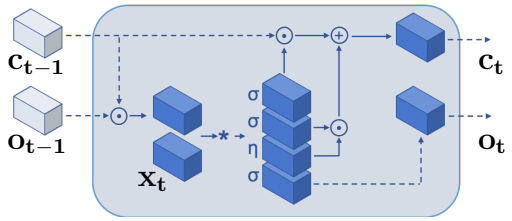
The proposed recurrent unit, LSTA, is illustrated in Fig. 4.1. The model is defined by the following equations,

$$\begin{aligned}
\nu_a &= \varsigma(\mathbf{x_t}, w_x) && (4.3)\\
(i_a, f_a, \mathbf{s_t}, a) &= (\sigma, \sigma, \sigma, \eta)(W_a * [\nu_a, \mathbf{s_{t-1}} \odot \eta(\mathbf{a_{t-1}})]) && (4.4)\\
\mathbf{a_t} &= f_a \odot \mathbf{a_{t-1}} + i_a \odot a && (4.5)\\
s &= \text{softmax}(\nu_a + \mathbf{s_t} \odot \eta(\mathbf{a_t})) && (4.6)\\
(i_c, f_c, c) &= (\sigma, \sigma, \eta)(W_c * [s \odot \mathbf{x_t}, \mathbf{o_{t-1}} \odot \eta(\mathbf{c_{t-1}})]) && (4.7)\\
\mathbf{c_t} &= f_c \odot \mathbf{c_{t-1}} + i_c \odot c && (4.8)\\
\nu_c &= \varsigma(\mathbf{c_t}, w_c + w_o \cdot \epsilon(s \odot \mathbf{x_t})) && (4.9)\\
\mathbf{o_t} &= \sigma(W_o * [\nu_c \odot \mathbf{c_t}, \mathbf{o_{t-1}} \odot \eta(\mathbf{c_{t-1}})]) && (4.10)
\end{aligned}$$

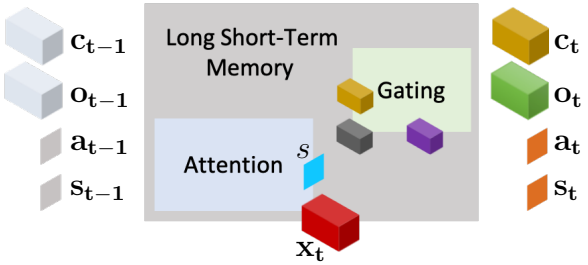
Eqs. 4.3-4.6 implement recurrent attention as detailed in Sec. 4.2.1, Eqs. 4.9-4.10 is the coupled output gating of Sec. 4.2.2. Bold symbols represent the recurrent variables: $(\mathbf{a_t}, \mathbf{s_t})$ of shape $N \times 1$, $(\mathbf{c_t}, \mathbf{o_t})$ of shape $N \times K$. Trainable parameters are: (W_a, W_c) are both K convolution kernels, (w_a, w_c) have shape $K \times C$, w_o has shape $C \times C$. N, K, C are introduced below. σ, η are sigmoid and tanh activation functions, $*$ is convolution, $\cdot$ is tensor product, $\odot$ is point-wise multiplication. ς, π are from the pooling model presented next. For a sequence prediction task, a hidden state $\mathbf{h}_t = \mathbf{o}_t \odot \eta(\mathbf{c}_t)$ can be exposed at each iteration.

4.2.1 Attention Pooling

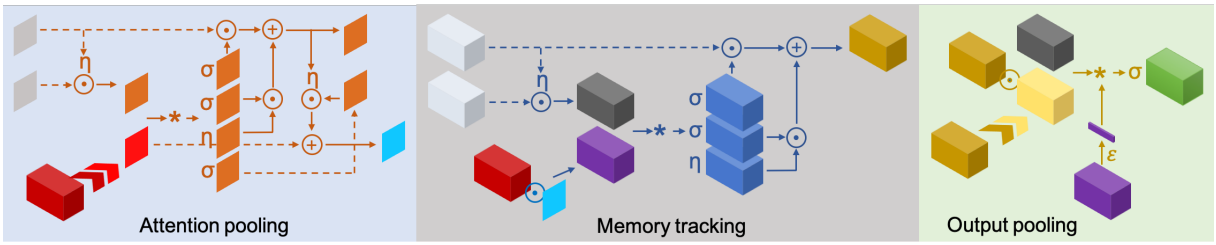
Given a matrix view $\mathbf{x}_{ik}$ of convolutional feature tensor $\mathbf{x}$ where i indexes one of N spatial locations and k indexes one of K feature planes, the



(a) LSTM unit



(b) LSTA unit



(c) LSTA internal blocks

Figure 4.1: (a) and (b) shows the high level block diagram of LSTM and LSTA. LSTA is obtained by incorporating attention to the input and updating the output gating. (c) shows the detailed internal diagram of LSTA. LSTA enhances LSTM with built-in attention (blue block) and flexible output gating (green block). Standard recurrence (gray block, Eqs. 4.7-4.8) tracks a memory state $\mathbf{c_t}$ using spatially attended features $s \odot \mathbf{x_t}$ and a gated view of previous memory $\mathbf{o_{t-1}} \odot \eta(\mathbf{c_{t-1}})$. To compute s (Eq. 4.6) we introduce a second internal state $\mathbf{a_t}$ to track attention (Eqs. 4.4-4.5) with the pooling scheme of Sec. 4.2.1 (Eq. 4.3). We use pooling (Eqs. 4.9) again to expose the output gate (Eqs. 4.10) in Sec. 4.2.2.

proposed approach aims at suppressing those activations $\mathbf{x}_i$ that are uncorrelated with the recognition task. That is, a $\varsigma(\mathbf{x}, w)$ of shape $1 \times N$ is required such that parameters w can be tuned in a way that $\varsigma(\mathbf{x}, w) \odot \mathbf{x}$ are the discriminative features for recognition. For egocentric activity recognition these can be from objects, hands, or implicit patterns representing object-hand interactions during manipulation.

The design of $\varsigma(\mathbf{x}, w)$ is grounded on the assumption that there is a limited number of pattern categories that are relevant for an activity recog-

nition task. Each category itself can, however, instantiate patterns with high variability during and across executions. Therefore ς should be selected from a pool of category-specific mappings, based on the current input $\mathbf{x}$. It is also required that both the selector and the pool of mappings be learnable and self-consistent, and realized with fewer tunable parameters.

A selector with parameters w maps an image features $\mathbf{x}$ into a category-score space $\mathcal{C}$ from which the category $c^* \in \mathcal{C}$ obtaining the highest score is returned. The proposed selector is of the form $c^* = \arg\max_c \pi(\epsilon(\mathbf{x}), \theta_c)$ where ϵ is a reduction and $\theta_c \in w$ are the parameters for scoring $\mathbf{x}$ against category c. If π is chosen to be equivariant to reduction ϵ then $\pi(\epsilon(\mathbf{x}), \theta_c) = \epsilon(\pi(\mathbf{x}, \theta_c))$ and $\{\epsilon^\perp(\pi(\cdot, \theta_c)), c \in \mathcal{C}\}$ can be used as the pool of category-specific mappings associated to ϵ. Here $\epsilon^\perp$ denotes the ϵ-orthogonal reduction, *e.g.*if ϵ is max-pooling along one dimension then $\epsilon^\perp$ is max-pooling along the other dimensions. That is, the proposed pooling model is determined by the triplet

$$(\varsigma) = (\epsilon, \pi, \{\theta_c\}) \ , \quad \pi \text{ is } \epsilon\text{-equivariant} \tag{4.11}$$

and realized on a feature tensor $\mathbf{x}$ by

$$\begin{aligned} \varsigma(\mathbf{x}, \{\theta_c\}) &= \epsilon^\perp(\pi(\mathbf{x}, \theta_{c^*})) && (4.12) \\ \text{where } c^* &= \arg\max_c \pi(\epsilon(\mathbf{x}), \theta_c) && (4.13) \end{aligned}$$

The proposed model uses

$$\begin{aligned} \epsilon(\mathbf{x}) &\leftarrow \text{spatial average pooling} \\ \pi(\epsilon, \theta_c) &\leftarrow \text{linear mapping} \end{aligned}$$

so $\varsigma(\mathbf{x}, \{\theta_c\})$ is a differentiable spatial mapping, *i.e.*, ς can be used as a trainable attention model for $\mathbf{x}$. This is related to class activation mapping [86] introduced for discriminative localization. Note however that, in contrast to [86] that uses strong supervision to train the selector directly,

the proposed approach leverages video-level annotation to implicitly learn an attention mechanism for video classification. The formulation of the proposed model is also a generalization: other choices are possible for the reduction ϵ, and the use of differentiable structured layers [117] in this context are an interesting direction for future work.

To inflate attention in LSTA, a new state tensor $\mathbf{a_t}$ of shape $N \times 1$ is introduced. Its update rule is that of standard LSTM (Eq. 4.5) with gatings $(f_a, i_a, \mathbf{s_t})$ and innovation a computed from the pooled $\nu_a = \varsigma(\mathbf{x_t}, w_a)$ as input (Eq. 4.4). The attention tensor s is computed using the hidden state $\mathbf{s_t} \odot \eta(\mathbf{a_t})$ as residual (Eq. 4.6), followed by a softmax calibration. Eqs. 4.7-4.10 implement the LSTA memory update based on the filtered input $s \odot \mathbf{x_t}$, this is described next.

4.2.2 Output Pooling

By analyzing the standard LSTM Eq. 4.2 with input $s \odot \mathbf{x_t}$ instead of $\mathbf{x_t}$, it becomes evident that $\mathbf{o_{t-1}}$ (output gating) has on $\eta(\mathbf{c_{t-1}})$ a same effect as s (attention) has on $\mathbf{x_t}$. Indeed, in Eq. 4.7 the gatings and innovation are all computed from $[s \odot \mathbf{x_t}, \mathbf{o_{t-1}} \odot \eta(\mathbf{c_{t-1}})]$. LSTA is built upon this analogy to enhance its output gating capacity and, consequently, its forget-update behavior of memory tracking.

For this, attention pooling is introduced in the output gating update. Instead of computing $\mathbf{o_t}$ as by Eq. 4.2 $s \odot \mathbf{x_t}$ is replaced with $\nu_c \odot \mathbf{c_t}$ to obtain update Eqs. 4.9-4.10, that is

$$\begin{aligned} \sigma(W_o * [s \odot \mathbf{x_t}, \mathbf{o_{t-1}} \odot \eta(\mathbf{c_{t-1}})]) &\leftarrow \text{standard gating} \\ \sigma(W_o * [\nu_c \odot \mathbf{c_t}, \mathbf{o_{t-1}} \odot \eta(\mathbf{c_{t-1}})]) & \\ \text{with } \nu_c = \varsigma(\mathbf{c_t}, w_c + w_o \cdot \epsilon(s \odot \mathbf{x_t})) &\leftarrow \text{output pooling} \end{aligned}$$

This choice is motivated as follows. In order to preserve the recursive nature of output gating, we need to keep on right-concatenating $\mathbf{o_{t-1}} \odot$

$\eta(\mathbf{c_{t-1}})$ to obtain the $2N \times K$-shaped tensor to convolve and tanh pointwise. Since the new memory state $\mathbf{c_t}$ is available at this stage, which already integrates $s \odot \mathbf{x_t}$, this can be used for left-concatenating instead of the raw attention-pooled input tensor. A filtered version of $\mathbf{c_t}$, $\nu_c \odot \mathbf{c_t}$, can also be produced introducing a second attention pooling neuron for localizing the actual discriminative memory component of $\mathbf{c_t}$, that is via ν_c, Eq. 4.9. Note that $\mathbf{c_t}$ integrates information from past memory updates by design, so localizing current activations this is pretty much required here. Consequently, and in contrast to feature tensors $\mathbf{x}$, the memory activations might not be well localized spatially. Thus slightly different version of Eq. 4.12 is used for output pooling, by removing $\epsilon^{\perp}$ to obtain a full-rank $N \times K$-shaped attention tensor ν_c.

To further enhance active memory localization, $s \odot \mathbf{x_t}$ is used to control the bias term of attention pooling, Eq. 4.9. This is done by applying a reduction $\epsilon(s \odot \mathbf{x_t})$ followed by a linear regression with learnable parameters w_o to obtain the instance-specific bias $w_o \cdot \epsilon(s \odot \mathbf{x_t})$ for activation mapping. Note that ϵ is the reduction associated to ς so this is consistent. A similar idea is proposed in Sec. 4.3.2.3 for cross-modal fusion in two-stream architecture. Results of the ablation study presented in Sec. 4.3.3.4 confirms that this further coupling of $\mathbf{c_t}$ with $\mathbf{x_t}$ boosts the memory distillation in the LSTA recursion, and consequently its tracking capability, by a significant margin.

4.3 LSTA for Egocentric Activity Recognition

Here, the problem of identifying fine-grained egocentric activities from trimmed videos is considered. This is a comparatively difficult task considered to action recognition since the activity class depends on the action and the object on to which the action is applied to. This requires the devel-

opment of a method that can simultaneously recognize the action as well as the object. In addition, the presence of strong ego-motion caused by the sharp movements of the camera wearer introduces noise to the video that complicates the encoding of motion in the video frame. While incorporating object detection can help the task of egocentric action recognition, still this would require fine-grain frame level annotations, becoming costly and impractical in a large scale setup.

4.3.1 Related Works

The most relevant deep learning methods for addressing egocentric vision problems are discussed in this section.

4.3.1.1 First Person Action Recognition

The works of [60, 59, 118] train specialized CNN for hand segmentation and object localization related to the activities to be recognized. These methods base on specialized pre-training for hand segmentation and object detection networks, requiring high amounts of annotated data for that purpose. Additionally, they just base on single RGB images for encoding appearance without considering temporal information. In [58, 119] features are extracted from a series of frames to perform temporal pooling with different operations, including max pooling, sum pooling, or histogram of gradients. Then, a temporal pyramid structure allows the encoding of both long term and short term characteristics. However, all these methods do not take into consideration the temporal order of the frames. Techniques that use a recurrent neural network such as LSTM [120, 121] and ConvLSTM [99, 103] are proposed to encode the temporal order of features extracted from a sequence of frames. Sigurdsson *et al.*[122] proposes a triplet network to develop a joint representation of paired third person and first person videos. Their method can be used for transferring knowledge

from third person domain to first person domain thereby partially solving the problem of lack of large first person datasets. Tang *et al.*[90, 123] add an additional stream that accepts depth maps to the two stream network for first person action recognition, thereby enabling the network to encode 3D information present in the scene. Li *et al.*[93] propose a deep neural network to jointly predict the gaze and action from first person videos, which requires gaze information during training.

Majority of the state-of-the-art techniques rely on additional annotations such as hand segmentation, object bounding box or gaze information. This allows the network to concentrate on the relevant regions in the frame and helps in distinguishing each activity from one another better. However, manually annotating all the frames of a video with these information is impractical. For this reason, development of techniques that can identify the relevant regions of a frame without using additional annotations is crucial.

4.3.1.2 Attention

Attention mechanism was proposed for focusing attention on features that are relevant for the task to be recognized. This includes [103, 93, 124] for first person action recognition, [125, 126, 127] for image and video captioning and [128, 125, 129] for visual question answering. The works of [98, 102, 130, 103, 105, 93] use an attention mechanism for weighting spatial regions that are representative for a particular task. Sharma *et al.*[98] and Zhang *et al.*[105] generate attention masks implicitly by training the network with video labels. Authors of [102, 130, 103] use top-down attention generated from the prior information encoded in a CNN pre-trained for object recognition while [93] uses gaze information for generating attention. The work of [131, 124] uses attention for weighting relevant frames, thereby adding temporal attention. This is based on the idea that not

all frames present in a video are equally important for understanding the action being carried out. In [131] a series of temporal attention filters is learnt that weight frame level features depending on their relevance for identifying actions. [124] uses change in gaze for generating the temporal attention. [104, 107] apply attention on both spatial and temporal dimensions to select relevant frames and the regions present in them.

Most of the existing techniques for generating spatial attention in videos consider each frame independently. Since videos are of sequential nature and have an absolute temporal consistency, per frame processing results in the loss of valuable information.

4.3.1.3 Relation to state-of-the-art alternatives

The proposed LSTA method generates the spatial attention map in a top-down fashion utilizing prior information encoded in a CNN pre-trained for object recognition and another pre-trained for action recognition. [103] proposes a similar top-down attention mechanism, which was presented in the previous chapter. However, in that approach the attention map is generated independently in each frame whereas in the proposed approach, the attention map is generated in a sequential manner. This is achieved by propagating the attention map generated from past frames across time by maintaining an internal state for attention. The proposed method uses attention on the motion stream followed by a cross-modal fusion of the appearance and motion streams, thereby enabling both streams to interact earlier in the layers to facilitate flow of information between them. [105] proposes an attention mechanism that takes into consideration the inputs from past frames. Their method is based on bottom-up attention and generates a single weight matrix which is trained with the video level label. However, the proposed method generates attention, based on the input, from a pool of attention maps which are learned using video level label

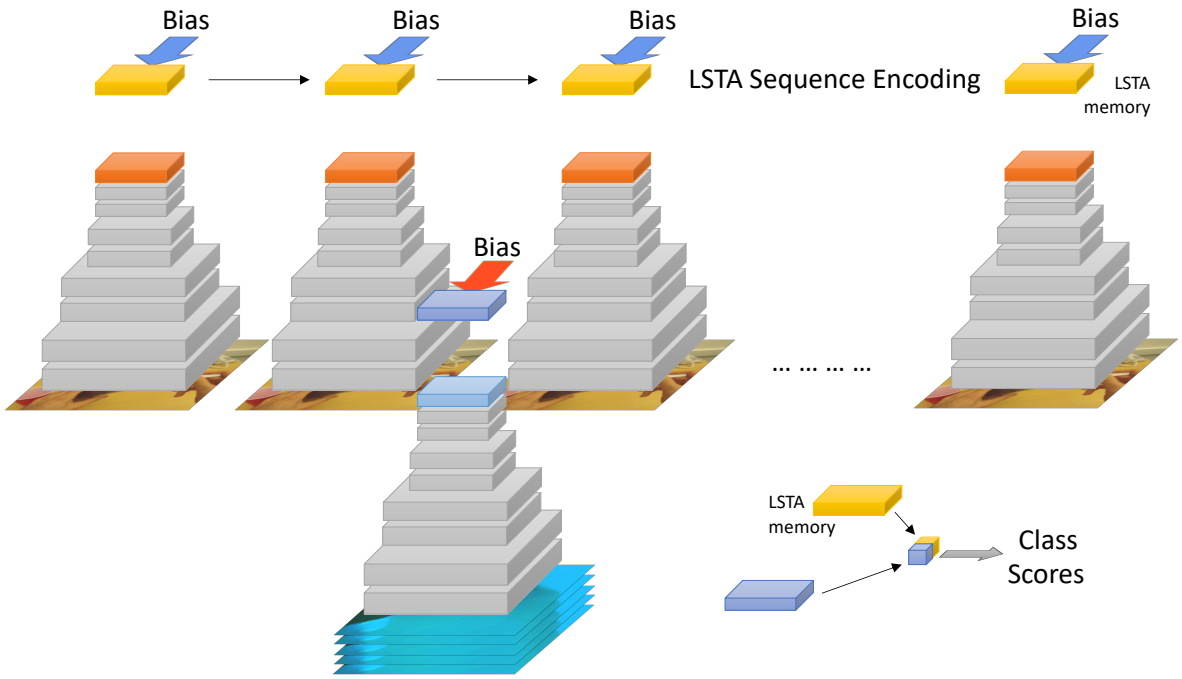


Figure 4.2: Cross-modal fusion in the proposed two stream architecture: proposed approach uses the stack-of-flow feature tensor to control LSTA bias, and the sequence of RGB feature tensors to control the bias of a ConvLSTM on flow features. ResNet-34 is used for feature extraction in both modalities, and perform standard late fusion for video-level prediction.

alone.

4.3.2 Proposed Method

This section explains the network architecture for egocentric activity recognition incorporating the LSTA module of Sec. 4.2. Like the majority of the deep learning methods proposed for action recognition, the proposed method also follows the two stream architecture; one stream for encoding appearance information from RGB frames and the second stream for encoding motion information from optical flow stacks.

4.3.2.1 Attention on Appearance Stream

The network consists of a ResNet-34 pre-trained on ImageNet for image recognition. The output of the final convolutional layer in layer 4 is used as the input of the LSTA module. From this frame level features, LSTA generates the attention map which is used to weight the input features. The depth of LSTA memory is selected as 512 and all the gates use a kernel size of 3×3.

A two stage training strategy is adopted for training the network. In the first stage, the FC layer in the classifier and the LSTA modules are trained while in the second stage, the convolutional layers in the layer 4 and the FC layer of ResNet-34 along with the layers trained in stage 1 are trained.

4.3.2.2 Attention on Motion Stream

In the proposed approach, a network trained on optical flow stacks is used for explicit motion encoding. This is achieved by using a ResNet-34 trained for action verbs (`take`, `put`, `pour`, `open`, etc.) recognition using an optical flow stack of 5 frames. The weights in the input convolutional layer of an ImageNet pre-trained network is averaged and replicated by 10 times to initialize the input layer. This is analogous to the ImageNet pre-training done on the appearance stream. The network is then trained for activity recognition as follows. The action-pretrained ResNet-34 FC weights are used for the parameter initialization of attention pooling (Eqs. 4.12-4.13) on Layer-4 flow features. This attention map is then used to weight the features for classification. Since the activities are temporally located in the videos and they are not sequential in nature, the optical flow corresponding to the five frames located in the temporal center of the videos are used as input.

4.3.2.3 Cross-modal Fusion

Majority of the existing methods with two stream architecture perform a simple late fusion by averaging for combining the outputs from the appearance and motion streams [48, 88]. Feichtenhofer *et al.* [132] propose a convolutional pooling strategy at the output of the final convolutional layer for improved fusion of the two streams. In [96] the authors observe that adding a residual connection from the motion stream to the appearance stream enables the network to improve the joint modeling of the information flowing through the two streams. Inspired by the aforementioned observations, a novel cross-modal fusion strategy is proposed in the earlier layers of the network in order to facilitate the flow of information across the two modalities thereby improving the learnt video representation.

Fig. 4.2 illustrates the proposed cross-modal fusion architecture for two stream networks. The appearance and motion streams are as described in Secs. 4.3.2.1 and 4.3.2.2. In this architecture, each stream is used to control the biases of other. The output of the motion stream is applied as bias to the gates of the LSTA layer. The output of the RGB CNN from all the input frames are applied to a 3D convolutional layer to obtain a summary feature of the input frames. A ConvLSTM is added in the motion stream as an embedding layer that is similar in functionality to the gates of the LSTA layer. The output of the 3D convolutional layer is then applied as bias to the gates of the ConvLSTM layer in the motion stream. In this way, each individual stream is made to influence the encoding of the other so that there is a flow of information between them deep inside the neural network. A late average fusion of the two individual streams' output is then performed to obtain the class scores.

4.3.3 Experiments and Results

4.3.3.1 Datasets

The proposed method is evaluated on four standard first person activity recognition datasets namely, GTEA 61 [87], GTEA 71 [87], EGTEA Gaze+ [93] and EPIC-KITCHENS [133]. GTEA 61 and GTEA 71 are relatively small scale datasets with 61 and 71 activity classes respectively. EGTEA Gaze+ is a recently developed large scale dataset with approximately 10K samples having 106 activity classes. EPIC-KITCHENS dataset is the largest egocentric activities dataset available now. The dataset consists of more than 28K video samples with 125 verb and 352 noun classes.

4.3.3.2 Implementation Details

The appearance and motion networks are first trained separately followed by a combined training of the two stream cross-modal fusion network. The networks are trained to minimize the cross-entropy loss. The appearance stream is trained for 200 epochs in stage 1 with a learning rate of 0.001 which is decayed after 25, 75 and 150 epochs at a rate of 0.1. In the second stage, the network is trained with a learning rate of 0.0001 for 100 epochs. The learning rate is decayed by 0.1 after 25 and 75 epochs. ADAM is used as the optimization algorithm. 25 frames uniformly sampled from the videos are used as input. The number of classes used in the output pooling (w_c in 4.2.2) is chosen as 100 for GTEA 61 and GTEA 71 datasets after empirical evaluation on the fixed split of GTEA 61. For EGTEA Gaze+ and EPIC-KITCHENS datasets, the value is scaled to 150 and 300 respectively, in accordance with the relative increase in the number of activity classes.

For the pre-training of the motion stream on action classification task, a learning rate of 0.01 is used which is reduced by 0.5 after 75, 150, 250

Ablation	**Accuracy (%)**
Baseline	51.72
Baseline + output pooling	62.07
Baseline + attention pooling	66.38
Baseline + pooling	68.1
LSTA	**74.14**
LSTA two stream late fusion	78.45
LSTA two stream cross-modal fusion	**79.31**

Table 4.1: Ablation analysis on GTEA 61 fixed split.

Method	Accuracy (%)		
	Activity	Action	Object
Baseline	51.72	65.52	57.76
Baseline+output pooling	62.07	79.31 (**+13.79**)	69.83 (+12,07)
Baseline+attention pooling	66.38	78.45 (+12,93)	74.14 (**+16,38**)
Baseline+pooling	68.10	79.31 (+13.79)	75.86 (+18,10)
LSTA	74.14	87.93 (**+22.41**)	79.31 (**+21,55**)

Table 4.2: Detailed ablation analysis on GTEA 61 fixed split. The action and object recognition score is computed by decomposing the action and objects from the predicted activity label.

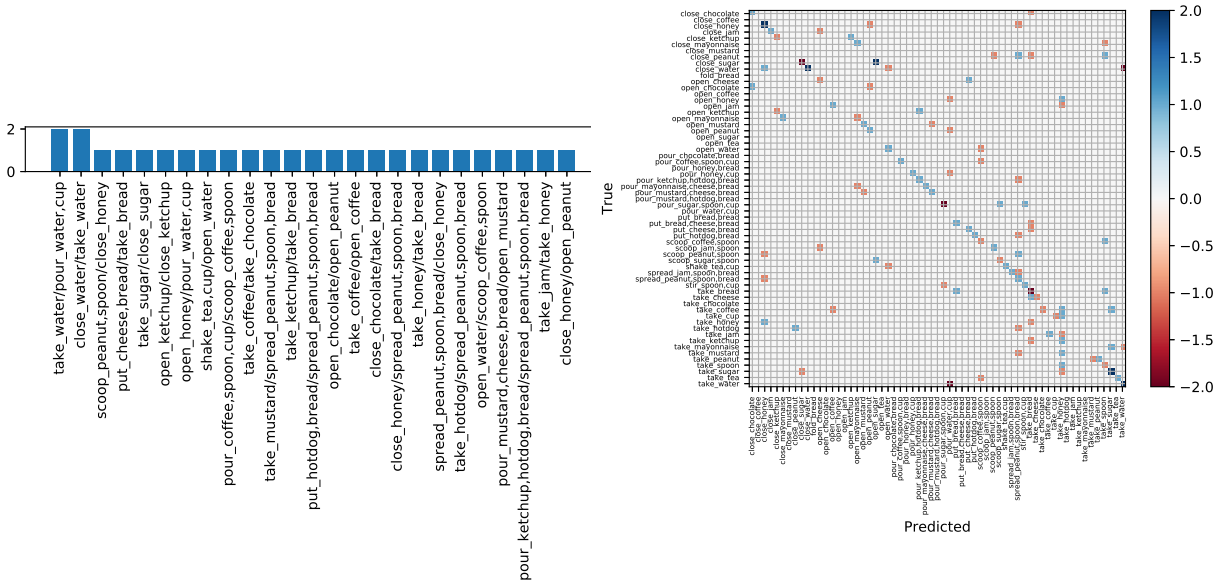
and 500 epochs and is trained for 700 epochs. In the activity classification stage, the network is trained for 500 epochs with a learning rate of 0.01. The learning rate is decayed after 50 and 100 epochs by 0.5. SGD algorithm is used for optimizing the parameter updates of the network.

The two stream network is trained for 200 epochs for GTEA 61 and GTEA 71 datasets while EGTEA Gaze+ is trained till 100 epochs, with a learning rate of 0.01 using ADAM algorithm. Learning rate is reduced by 0.99 after each epoch. A batch size of 32 is used for all networks. Random horizontal flipping and multi-scale corner cropping techniques proposed in [88] are used during training and the center crop of the frame is used during the evaluation stage.

4.3.3.3 Ablation Study

An extensive ablation analysis has been carried out, on the fixed split of GTEA 61 dataset, to determine the performance improvement obtained by each component of the proposed method. The results are shown in Tabs. 4.1 and 4.2. Tab. 4.1 compares the performance of RGB and two stream networks on the top and bottom sections respectively. Tab. 4.2 lists a breakdown of the recognition performance. For this, the action recognition and object recognition performance of a network trained for activity recognition is computed. There are some activity classes with multiple objects and these objects are combined to form a meta-object class for this analysis. Figs. 4.3 - 4.7 show details of the classes which are improved by the proposed LSTA variants over the baseline (ConvLSTM) and the difference of the confusion matrices. The top 25 improved classes are shown in the comparison graphs and those with less number list all the improved classes. The difference of confusion matrices show the overall details of the classes which are improved. Ideally, the positive values should be in the diagonal and the negative values off-diagonal.

A network with vanilla ConvLSTM is chosen as the baseline since LSTA without attention and output pooling converges to the standard ConvLSTM. The baseline model results in an activity recognition accuracy of 51.72%. The impact of each of the contributions explained in Sec. 4.2 is then analyzed. First, the effect of output pooling on the baseline is analyzed. By adding output pooling, the activity recognition performance is improved by 8% while a gain of 13.79% is achieved for action recognition. From Fig. 4.3 which compares the baseline (ConvLSTM) with a network having baseline+output pooling, it can be seen that adding output pooling to the ConvLSTM improves the network's capability in recognizing different actions with the same objects (`take_water/pour_water,cup`

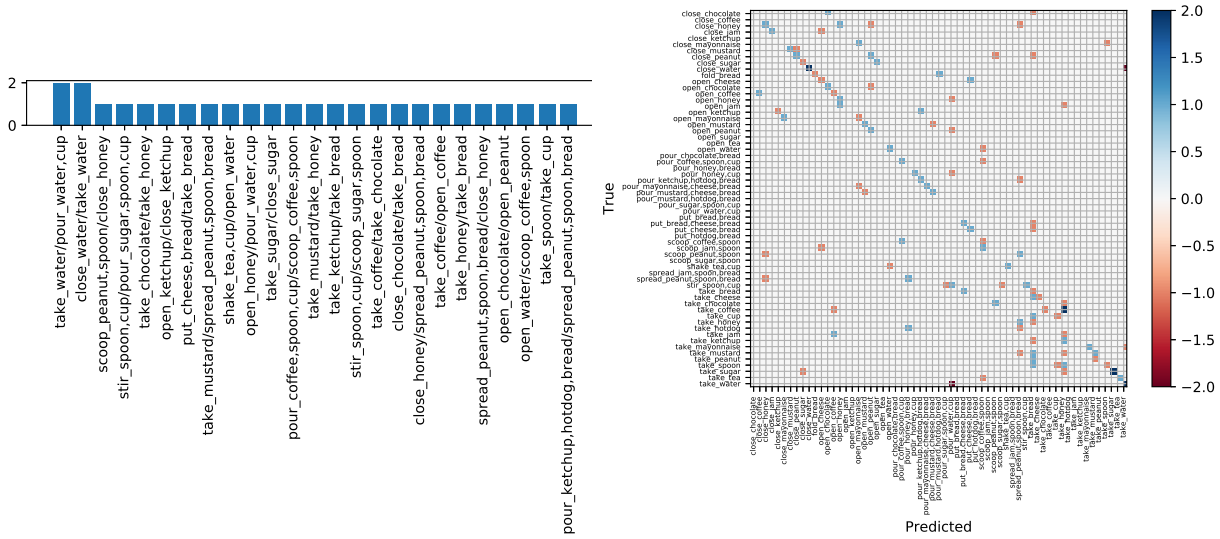


(a) (b)

Figure 4.3: (a) Most improvement categories by adding output pooling to the baseline on GTEA 61 fixed split. X axis labels are in the format true label (baseline + output pooling)/predicted label (baseline). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.

and `close_water/take_water`). This confirms that the output gating of LSTM affects memory tracking and replacing the output gating of LSTM with the proposed output pooling technique enables the network to propagate a filtered a version of the memory which is localized on the most discriminative components. This improves the tracking of relevant spatio-temporal patterns in the memory and consequently boosts recognition performance.

Adding attention pooling to the baseline improves the activity recognition performance by 14% and object recognition performance by 16.38%. Attention pooling enables the network to identify the relevant regions in the input frame and to maintain a history of the relevant regions seen in the past frames. This enables the network to have a smoother tracking of attentive regions. From Fig. 4.4, it can be seen that the network with

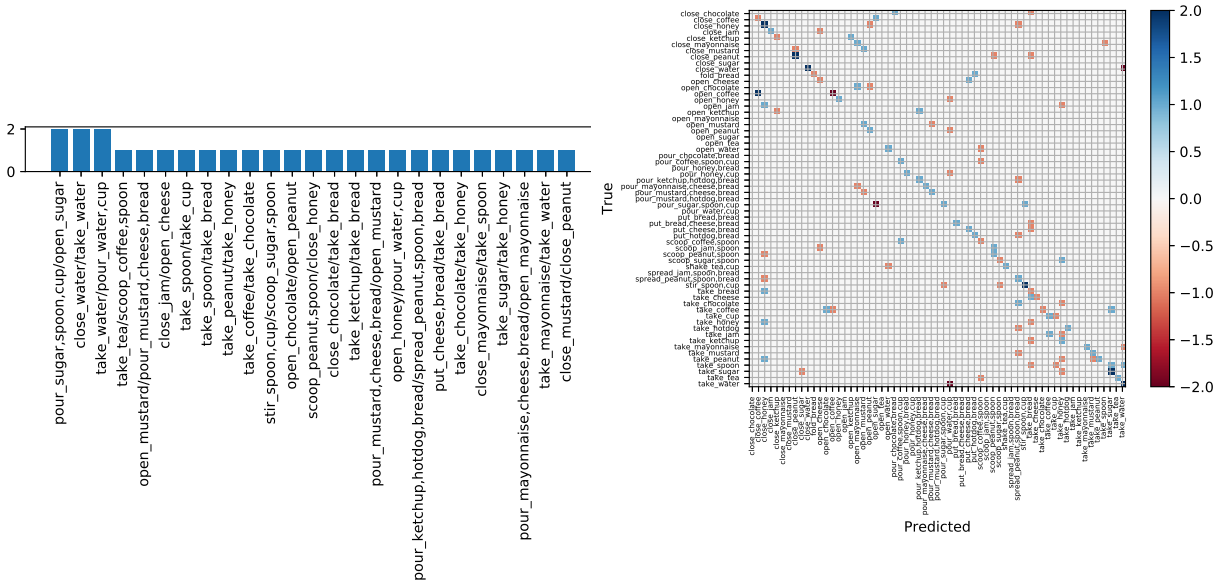


(a) (b)

Figure 4.4: (a) Most improvement categories by adding attention pooling to the baseline on GTEA 61 fixed split. X axis labels are in the format true label (baseline + attention pooling)/predicted label (baseline). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.

attention pooling described in Sec. 4.2 improves the categories with different actions and same objects as well as activity classes with multiple objects (`stir_spoon,cup/pour_sugar,spoon,cup`; `put_cheese,bread/take_bread`; `pour_coffee,spoon,cup/scoop_coffee,spoon`, etc.). It should be noted that this is equivalent to a network with two ConvLSTMs, one for attention tracking and one for frame level feature tracking.

Incorporating both attention and output pooling to the baseline results in a gain of 16% in activity recognition. By analyzing the top improved classes, as shown in Fig. 4.5, it can bee seen that the model has increased its capacity to correctly classify both actions and objects. Adding bias control, as explained in Sec. 4.2, results in the proposed LSTA model and gains an additional improvement of 6% in activity recognition accuracy. This further verifies the hypothesis in Sec. 4.2 that bias control increases



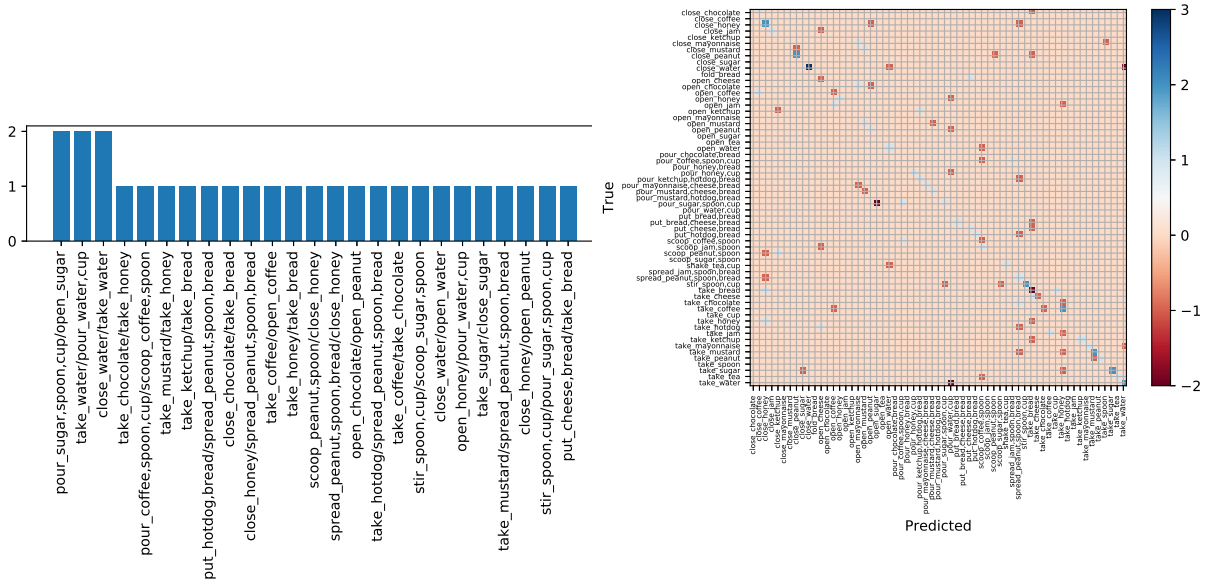
(a) (b)

Figure 4.5: (a) Most improvement categories by adding both attention and output pooling to the baseline on GTEA 61 fixed split. X axis labels are in the format true label (baseline + pooling)/predicted label (baseline). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.

the active memory localization of the network. This is also evident from Tab. 4.2 where an increase of 22.41% is obtained for action recognition.

It is worth noting that output pooling boosts action recognition performance more (+13.79% action vs +12,07% object) while with attention pooling the object recognition performance receives a higher gain (+12,93% vs +16,38%). Coupling attention and output pooling through bias control finally boosts performance by a significant margin on both (+22.41% vs +21,55%). This provides further evidence that the two contributions are complementary and reflects the intuitions behind the design choices of LSTA, making the improvements explainable and the benefits of each of the contributions are transparently confirmed by this analysis.

The performance improvement achieved by applying attention to the motion stream is also evaluated. The baseline is a ResNet-34 pre-trained

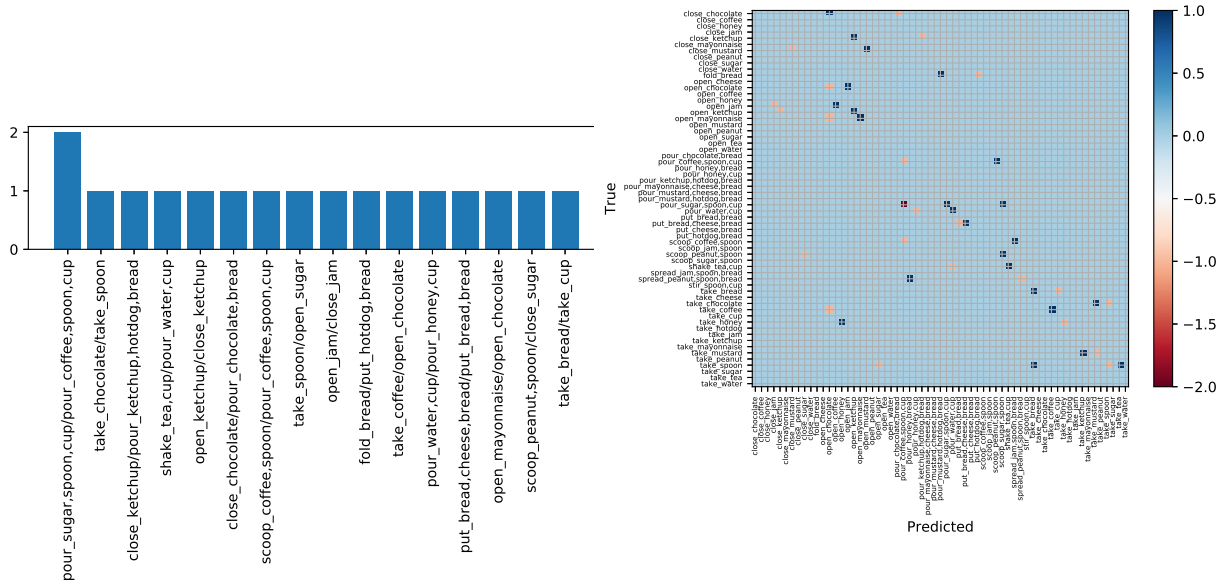


(a) (b)

Figure 4.6: (a) Most improvement categories by adding attention and output pooling with bias control (full LSTA model) to the baseline on GTEA 61 fixed split. X axis labels are in the format true label (LSTA)/predicted label (baseline). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.

on actions followed by training for activities. An accuracy of 40.52% is obtained for the network with attention compared to the 36.21% of the baseline. Fig. 4.10 (fourth row) visualizes the attention map generated by the network. For visualization, the resized attention map is overlayed on the RGB frames corresponding to the optical flow stack used as input. From the figure, it can be seen that the network generates the attention map around/near the hands, where the discriminant motion is occurring, thereby enabling the network to recognize the activity undertaken by the user. It can also be seen that the attention maps generated by the appearance stream and the flow stream are complementary to each other; appearance stream focuses on the object regions while the motion stream focuses on hand regions.

Next the performance improvement obtained by adding the cross-modal



(a) (b)

Figure 4.7: (a) Most improvement categories by two stream cross-modal fusion over two stream on GTEA 61 fixed split. X axis labels are in the format true label (two stream cross-modal fusion)/predicted label (two stream late fusion). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.

fusion technique explained in Sec. 4.3.2.3 over the traditional late fusion two stream approach is analyzed. The cross-modal fusion approach improves by 1% over late fusion. Fig. 4.7 shows that the cross-modal fusion approach is able to correctly identify activities with same objects. The fifth and sixth rows of Fig. 4.10 visualize the attention maps generated after two stream cross-modal fusion training. It can be seen that the motion stream attention expands to regions containing objects. This validates the effect of cross-modal fusion where the two networks are made to interact deep inside the network.

4.3.3.4 Comparative Analysis

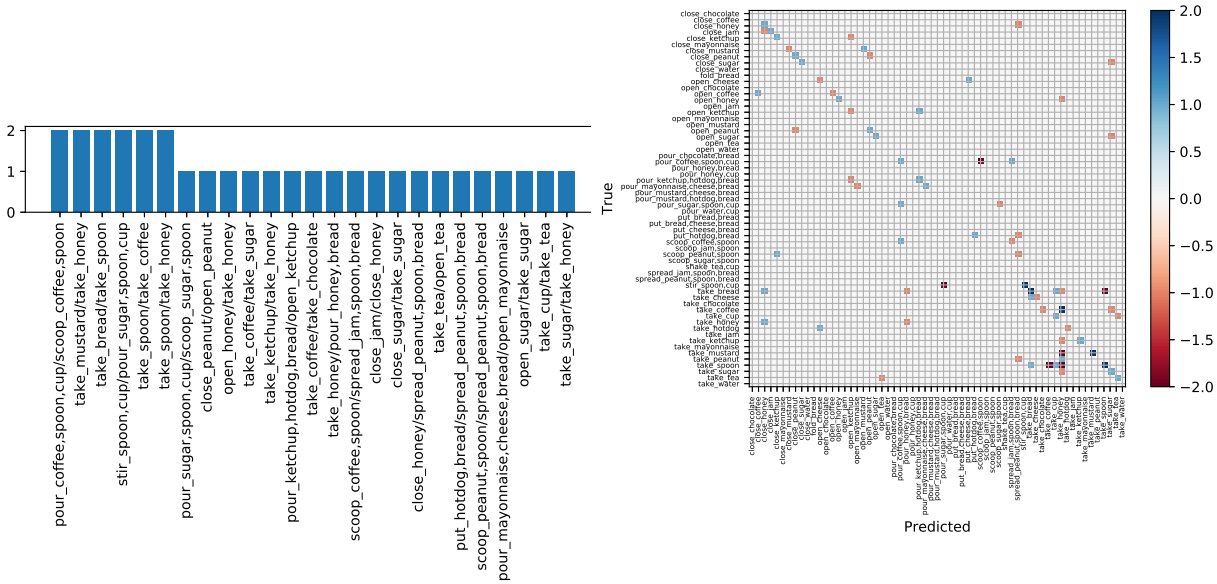
In this section, the performance of LSTA over two closely related methods, namely, eleGAtt [105] and ego-rnn [103] is compared. Results are shown

Method	**Accuracy (%)**
eleGAtt [105]	59.48
ego-rnn [103]	63.79
LSTA	**74.14**
ego-rnn two stream [103]	77.59
LSTA two stream	**79.31**

Table 4.3: Comparative analysis on GTEA 61 fixed split.

in Tab. 4.3. EleGAtt is an attention mechanism which can be applied to any generic RNN using its hidden state for generating the attention map. In this comparison, eleGAtt is evaluated on LSTM, consisting of 512 hidden units, with the same training setting as LSTA for, fair comparison. EleGAtt learns a single weight matrix for generating the attention map irrespective of the input whereas LSTA generates the attention map from a pool of weights which are selected in a top-down manner based on input. This enables the selection of a proper attention map for each input activity class. This leads to a performance gain of 13% over eleGAtt. From Fig. 4.8, analyzing the classes with the highest improvement by LSTA compared to eleGAtt reveals that eleGAtt fails in identifying the objects correctly (`take_mustard/take_honey`; `take_bread/take_spoon`; `take_spoon/take_honey`, etc.). This shows the attention map generated by LSTA selects more relevant regions compared to eleGAtt.

Ego-rnn [103], presented in the previous chapter, generates a per frame attention map which has no dependency on the information present in the previous frames. This can result in selecting different objects in adjacent frames. On the contrary, LSTA uses an attention memory to track the previous attention maps enabling their smooth tracking. This results in a 10% improvement obtained by LSTA over ego-rnn. Detailed analysis on the classification results show that ego-rnn struggles to classify activities involving multiple objects (Fig. 4.9). Since the attention map generated in



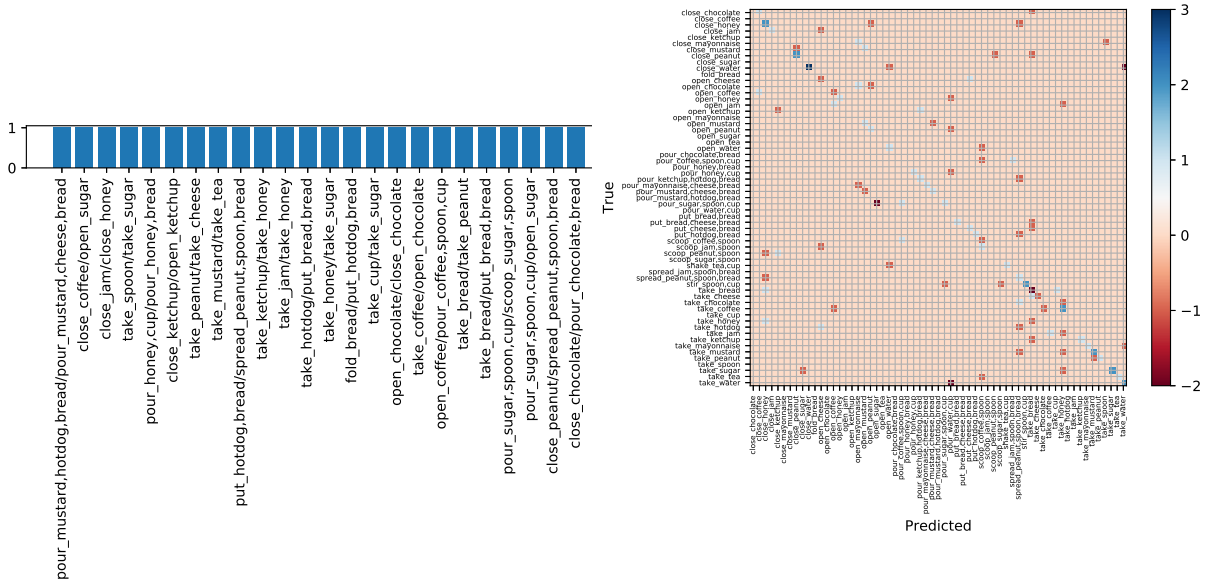
(a) (b)

Figure 4.8: (a) Most improvement categories by LSTA over eleGAtt-LSTM on GTEA 61 fixed split. X axis labels are in the format true label (LSTA)/predicted label (eleGAtt-LSTM). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.

each frame is independent of the previous frames, the network fails to track previously activated regions, thereby resulting in wrong predictions. This is further illustrated by visualizing the attention maps produced by ego-rnn and LSTA in Fig. 4.10. From the figure, one can see that ego-rnn (second row) fails to identify the relevant object in the case of `close chocolate` example and it failed to track the object in the final frames in the case of `scoop coffee` example. LSTA with cross-modal fusion performs 2% better than ego-rnn two stream.

4.3.3.5 State-of-the-art comparison

The proposed approach is compared against the state-of-the-art methods in Tab. 4.4. The methods listed in the first section of the table uses strong supervision signals such as gaze [56, 93], hand segmentation [60] or object



(a) (b)

Figure 4.9: (a) Most improvement categories by LSTA over ego-rnn on GTEA 61 fixed split. X axis labels are in the format true label (LSTA)/predicted label (ego-rnn). Y axis shows the number of corrected samples for each class. (b) shows the difference of confusion matrices.

bounding boxes [60] during the training stage. Two stream [48], I3D [89] and TSN [88] are methods proposed for action recognition from third person videos while all other methods except eleGAtt [105] are proposed for first-person activity recognition. eleGAtt [105] is proposed as a generic method for incorporating attention mechanism to any RNN modules. From the table, we can see that the proposed method outperforms all the existing methods for egocentric activity recognition.

4.3.3.6 EPIC-KITCHENS

In this dataset, the labels are provided in the form of `verb` and `noun`, which are combined to form an activity class. The fact that not all combinations of verbs and nouns are feasible and that not all test classes might have a representative training sample make it a challenging problem. The network

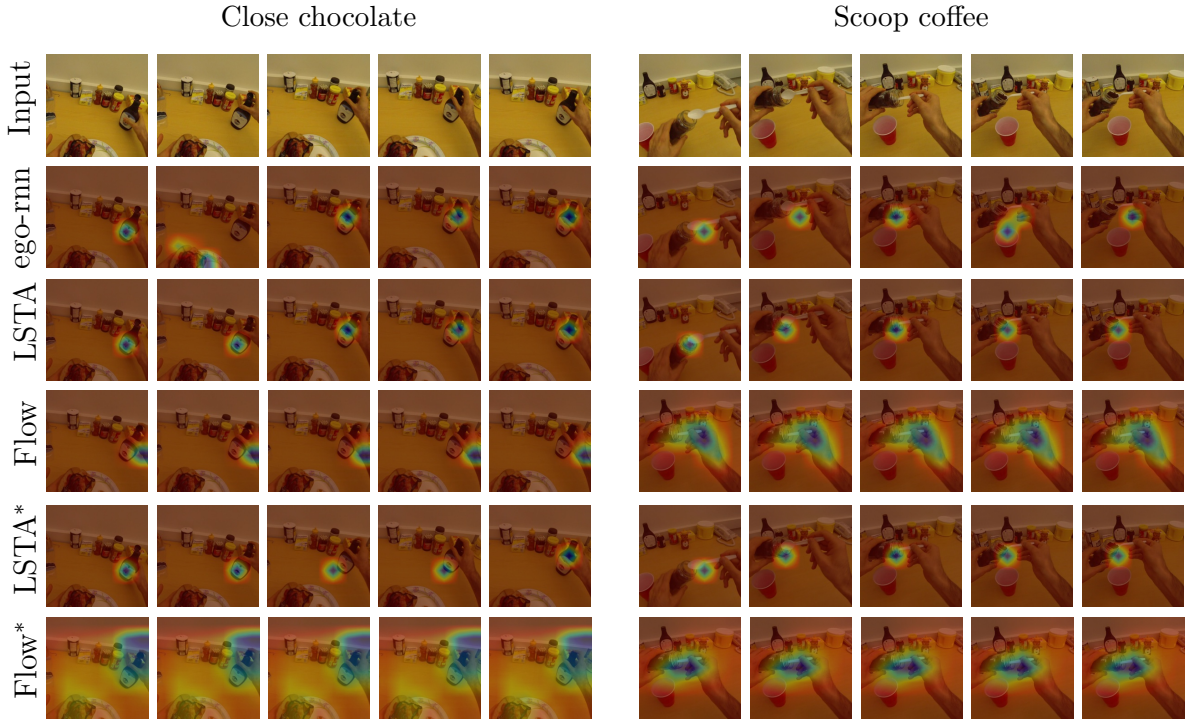


Figure 4.10: Attention maps generated by ego-rnn (second row) and LSTA (third) for two video sequences. The 5 frames that are uniformly sampled from the 25 frames used as input to the corresponding networks are shown. Fourth row shows the attention map generated by the motion stream. Fifth and sixth rows show the attention map generated by the appearance and flow streams after two stream cross-modal training. For flow, the attention map is visualized on the five frames corresponding to the optical flow stack given as input. (*: Attention map obtained after two stream cross-modal fusion training).

is trained for multi-task classification with `verb` and `noun` and `activity` supervision. The activity classifier activations are used to control the bias of `verb` and `noun` level classifiers. The dataset provides two evaluation settings, seen kitchens (S1) and unseen kitchens (S2). An accuracy of 30.33% (S1) and 16.33% (S2) using just RGB frames. The best performing baseline is a two stream TSN that achieves 20.54% (S1) and 10.89% (S2) [133]. The proposed model is particularly strong on `verb` prediction (59%) where a gain of +10% points is obtained over TSN. `verb` in this context is typically describing actions that develop into an activity over time, confirming once

Methods	Datasets			
	GTEA 61*	**GTEA 61**	**GTEA 71**	**EGTEA**
Li *et al.* [56]**	66.8	64	62.1	46.5
Ma *et al.* [60]**	75.08	73.02	73.24	-
Li *et al.* [93]**	-	-	-	53.3
Two stream [48]	57.64	51.58	49.65	41.84
I3D [89]	-	-	-	51.68
TSN [88]	67.76	69.33	67.23	55.93
eleGAtt [105]	59.48	66.77	60.83	57.01
ego-rnn [103]	77.59	79	77	60.76
LSTA-RGB	74.14	71.32	66.16	57.94
LSTA	**79.31**	**80.01**	**78.14**	**61.86**

Table 4.4: Comparison with state-of-the-art methods on popular egocentric datasets, recognition accuracy in % is reported. (*: fixed split; **: trained with strong supervision).

	Methods	Accuracy (%)		
		Verb	**Noun**	**Action**
S1	2SCNN (RGB)	40.44	30.46	13.67
	2SCNN (two stream)	42.16	29.14	13.23
	TSN (RGB)	45.68	36.80	19.86
	TSN (two stream)	48.23	36.71	20.54
	LSTA (RGB)	58.25	**38.93**	30.16
	LSTA (two stream)	**59.55**	38.35	**30.33**
S2	2SCNN (RGB)	34.89	21.82	10.11
	2SCNN (two stream)	36.16	18.03	7.31
	TSN (RGB)	34.89	21.82	10.11
	TSN (two stream)	39.4	22.7	10.89
	LSTA (RGB)	45.51	**23.46**	15.88
	LSTA (two stream)	**47.32**	22.16	**16.63**

Table 4.5: Comparison of recognition accuracies with state-of-the-art in EPIC-KITCHENS dataset.

more LSTA efficiently learns encoding of sequences with localized patterns.

4.4 Conclusion

This chapter presented LSTA that extends LSTM with two core features: 1) attention pooling that spatially filters the input sequence and 2) output pooling that exposes a distilled view of the memory at each iteration. As shown in a detailed ablation study, both contributions are essential for a smooth and focused tracking of a latent representation of the video to achieve superior performance in classification tasks where the discriminative features can be localized spatially. A novel CNN-LSTA architecture featuring a novel cross-modal fusion is then presented. The proposed model is evaluated on four egocentric activity recognition benchmarks and achieved state-of-the-art recognition accuracy.

Chapter 5

Conclusions and Future Work

Visual understanding from videos has been a long standing and well researched area in computer vision. This thesis addressed this research area in the context of video representation learning for action recognition. The three chapters investigated the open challenges associated with this task and presented solutions for addressing them. This final chapter summarizes the contributions of this thesis and discuss about future research directions.

5.1 Conclusions

The thesis first investigated the importance of using an order preserving encoding technique for aggregating the features from video frames. Analysis of popular CNN-RNN architectures revealed the significance of RNNs in aggregating frame level features generated by CNN architectures. Further analysis showed that an RNN module with convolutional gates, ConvLSTM, encodes the spatio-temporal features better than the traditional FC LSTM. The effectiveness of such an architecture was evaluated on two applications, violent video recognition and first person interaction recognition.

This was followed by a study for analyzing the importance of selective spatial encoding via attention, based on the idea of CAM. This lead to the

development of two end-to-end learnable neural architectures that leverage spatial attention for weighting the discriminant spatial regions present in the video frame. Detailed analysis on two video classification tasks, egocentric activity recognition and third-person action recognition, showed that the models were able to elicit spatial attention on relevant objects/regions present in the scene that enables the models to discriminate one action category from another. The proposed networks with spatial attention, when trained on video-level supervision alone were capable of shifting its attention from ImageNet prior to the corresponding task-specific activity relevant regions.

Inspired from the results of the first two studies, the next step was to integrate the two ideas, *i.e.*, recurrent convolutional encoding and spatially selective encoding. This is done by conducting a detailed analysis of Long Short-Term Memory (LSTM) design and improving it to accommodate the aforementioned features. This resulted in the development of Long Short-Term Attention (LSTA), a novel recurrent neural unit, with in built attention mechanism and an updated output gating compared to LSTM. The former allows the network to attend to the relevant regions of the video frame and maintain a smooth tracking of the attended regions. The latter enables the network to expose a distilled view of the internal memory which localizes the active memory components thereby improving the video representation generated. Detailed experimental analysis on popular egocentric activity recognition benchmarks revealed the significance of these two innovations in generating an effective video representation.

5.2 Future Work

The problem of understanding visual content in videos is not only limited to simply predicting the class category of the action present in the video.

As mentioned in chapter 1, action recognition is only a subset of the video content understanding problem. In the real world scenario, videos contain a wide range of variations such as with multiple persons carrying out similar or different activities, with multiple activities that can occur either simultaneously or sequentially and so on. The thesis investigated techniques for generating effective video representation based on deep learning and use it for predicting the class category of the action. However, many real life applications desire a more flexible representation of the video. For instance, systems that can generate a textual or speech description of a video are of significant relevance. On the other hand, a technique that can retrieve a video or a portion of a long video based on a query sentence is also very valuable as it allows for more diversified search and in enhancing the user experience. Development of such systems, which require the successful integration of multiple streams of research areas involving computer vision, natural language processing, speech recognition, knowledge representation, etc., is the path to follow in order to achieve true Artificial Intelligence. A brief outline of the path to follow in order to achieve this ultimate goal can be summarised as:

- To perform a thorough evaluation of the proposed LSTA recurrent neural unit on more diverse set of datasets such as Something-Something dataset [134], Charades dataset [135] and Moments in Time dataset [136], etc. that require strong temporal reasoning in order to correctly classify the action categories present. This will help us to understand the generalization capability of the proposed method and to build on it further to address the current limitations emerging from this analysis. One possible approach is to integrate VLAD as an intermediate differentiable layer, thus joining the distinguishing features of both methods. This could lead to the development of techniques capable of distilling structured information that characterize the evolution of

action-discriminant spatio-temporal features in videos.

- Since videos in the wild are not temporally trimmed and are of longer duration, techniques have to be developed in order to temporally localize the action sequence of interest. This can be achieved by the application of temporal attention. Temporal attention enables the network to focus on those frames which contain relevant information while discarding the irrelevant frames. This can allow the network to temporally localize small sequences of interest as well as in identifying action sequences of longer temporal duration.

- With the development of effective deep learning techniques for addressing natural language problems such as language translation, speech recognition, sentiment analysis, etc., integration of video and textual modalities became less cumbersome. Such an integrated system can be trained end-to-end, provided there is big enough annotated data. Another advantage of integrating different modalities is that the system could be trained in an unsupervised manner. For instance, majority of the videos shared in social media contain information in the form of captions and comments. This additional modality of information can be exploited to train a network without providing extra annotations in the form of labels and can achieve a high degree of generalization because of the huge amount of data available in social media and video streaming websites.

Bibliography

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proc. CVPR*, 2016.

[2] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, 2017.

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proc. ECCV*, 2016.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. ICCV*, 2017.

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. CVPR*, 2016.

[6] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely Connected Convolutional Networks. In *Proc. CVPR*, 2017.

[7] M Russell Harter and Cheryl J Aine. Brain mechanisms of visual selective attention. *Varieties of attention*, pages 293–321, 1984.

[8] Sabine Kastner Ungerleider and Leslie G. Mechanisms of Visual Attention in the Human Cortex. *Annual Review of Neuroscience*, 23(1):315–341, 2000.

[9] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jrgen Schmidhuber. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

[12] Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. Translation modeling with bidirectional recurrent neural networks. In *Proc. Conference on Empirical Methods in Natural Language Processing*, 2014.

[13] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional LSTM. In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013.

[14] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Proc. NIPS*, 2015.

[15] Bruno Stuner, Clément Chatelain, and Thierry Paquet. Handwriting recognition using cohort of lstm and lexicon verification with extremely large lexicon. *arXiv preprint arXiv:1612.07528*, 2016.

[16] Jianshu Zhang, Jun Du, and Lirong Dai. A gru-based encoder-decoder approach with attention for online handwritten mathematical expression recognition. In *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, 2017.

[17] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Pro. CVPR*, 2015.

[18] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. In *Proc. NIPS*, 2015.

[19] Viorica Pătrăucean, Ankur Handa, and Roberto Cipolla. Spatio-temporal video autoencoder with differentiable memory. In *Proc. ICLR Workshop*, 2016.

[20] Jefferson Ryan Medel and Andreas Savakis. Anomaly Detection in Video Using Predictive Convolutional Long Short-Term Memory Networks. *arXiv preprint arXiv:1612.00390*, 2016.

[21] Enrique Bermejo Nievas, Oscar Deniz Suarez, Gloria Bueno García, and Rahul Sukthankar. Violence detection in video using computer vision techniques. In *International Conference on Computer Analysis of Images and Patterns*, 2011.

[22] Piotr Bilinski and Francois Bremond. Human violence recognition and detection in surveillance videos. In *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2016.

[23] Silvia Pfeiffer, Stephan Fischer, and Wolfgang Effelsberg. Automatic audio content analysis. In *ACM International Conference on Multimedia*, 1997.

[24] Theodoros Giannakopoulos, Aggelos Pikrakis, and Sergios Theodoridis. A multi-class audio classification method with respect to violent content in movies using bayesian networks. In *IEEE Workshop on Multimedia Signal Processing (MMSP)*, 2007.

[25] Wojtek Zajdel, Johannes D Krijnders, Tjeerd Andringa, and Dariu M Gavrila. CASSANDRA: audio-video sensor fusion for aggression detection. In *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2007.

[26] Esra Acar, Frank Hopfgartner, and Sahin Albayrak. Breaking down violence detection: Combining divide-et-impera and coarse-to-fine strategies. *Neurocomputing*, 208:225–237, 2016.

[27] Nuno Vasconcelos and Andrew Lippman. Towards semantically meaningful feature spaces for the characterization of video content. In *ICIP*, 1997.

[28] C Clarin, J Dionisio, and M Echavez. DOVE: Detection of movie violence using motion intensity analysis on skin and blood. Technical report, University of the Philippines, 01 2005.

[29] Liang-Hua Chen, Hsi-Wen Hsu, Li-Yun Wang, and Chih-Wen Su. Violence detection in movies. In *International Conference on Computer Graphics, Imaging and Visualization (CGIV)*, 2011.

[30] Oscar Deniz, Ismael Serrano, Gloria Bueno, and Tae-Kyun Kim. Fast violence detection in video. In *Proc. International Conference on Computer Vision Theory and Applications (VISAPP)*, 2014.

[31] Ankur Datta, Mubarak Shah, and N Da Vitoria Lobo. Person-on-person violence detection in video data. In *ICPR*, 2002.

[32] Datong Chen, Howard Wactlar, Ming-yu Chen, Can Gao, Ashok Bharucha, and Alex Hauptmann. Recognition of aggressive human behavior using binary local motion descriptors. In *International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS)*, 2008.

[33] Fillipe DM De Souza, Guillermo C Chavez, Eduardo A do Valle Jr, and Arnaldo de A Araújo. Violence detection in video using spatio-temporal features. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, 2010.

[34] Tal Hassner, Yossi Itcher, and Orit Kliper-Gross. Violent Flows: Real-Time Detection of Violent Crowd Behavior. In *Proc. CVPR Workshops*, June 2012.

[35] Long Xu, Chen Gong, Jie Yang, Qiang Wu, and Lixiu Yao. Violent video detection based on MoSIFT feature and sparse coding. In *Proc. ICASSP*, 2014.

[36] Sadegh Mohammadi, Hamed Kiani, Alessandro Perina, and Vittorio Murino. Violence detection in crowded scenes using substantial derivative. In *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2015.

[37] Paolo Rota, Nicola Conci, Nicu Sebe, and James M Rehg. Real-life violent social interaction detection. In *ICIP*, 2015.

[38] Ismael Serrano Gracia, Oscar Deniz Suarez, Gloria Bueno Garcia, and Tae-Kyun Kim. Fast fight detection. *PloS one*, 10(4):e0120448, 2015.

[39] Yuan Gao, Hong Liu, Xiaohu Sun, Can Wang, and Yi Liu. Violence detection using Oriented VIolent Flows. *Image and Vision Computing*, 48:37–41, 2016.

[40] Tao Zhang, Wenjing Jia, Xiangjian He, and Jie Yang. Discriminative Dictionary Learning With Motion Weber Local Descriptor for Violence Detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(3):696–709, 2017.

[41] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Proc. NIPS*, 2014.

[42] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional LSTM. In *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013.

[43] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron C Courville, Ruslan Salakhutdinov, Richard S Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *Proc. ICML*, 2015.

[44] Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence-video to text. In *Proc. ICCV*, 2015.

[45] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised Learning of Video Representations using LSTMs. In *Proc. ICML*, 2015.

[46] Zhihong Dong, Jie Qin, and Yunhong Wang. Multi-stream Deep Networks for Person to Person Violence Detection in Videos. In *Chinese Conference on Pattern Recognition*, 2016.

[47] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012.

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

[49] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016.

[50] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 2012.

[51] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, 2010.

[52] Alireza Fathi, Ali Farhadi, and James M Rehg. Understanding Egocentric Activities. In *Proc. ICCV*, 2011.

[53] Tomas McCandless and Kristen Grauman. Object-Centric Spatio-Temporal Pyramids for Egocentric Activity Recognition. In *Proc. BMVC*, 2013.

[54] Alireza Fathi and James M Rehg. Modeling Actions through State Changes. In *Proc. CVPR*, 2013.

[55] Kenji Matsuo, Kentaro Yamada, Satoshi Ueno, and Sei Naito. An Attention-Based Activity Recognition for Egocentric Video. In *Proc. CVPR Workshops*, 2014.

[56] Yin Li, Zhefan Ye, and James M Rehg. Delving into Egocentric Actions. In *Proc. CVPR*, 2015.

[57] Lorenzo Baraldi, Francesco Paci, Giuseppe Serra, Luca Benini, and Rita Cucchiara. Gesture Recognition in Ego-centric Videos Using Dense Trajectories and Hand Segmentation. In *CVPR Workshops*, 2014.

[58] Michael S Ryoo, Brandon Rothrock, and Larry Matthies. Pooled Motion Features for First-Person Videos. In *Proc. CVPR*, 2015.

[59] Suriya Singh, Chetan Arora, and CV Jawahar. First Person Action Recognition Using Deep Learned Descriptors. In *Proc. CVPR*, 2016.

[60] Minghuang Ma, Haoqi Fan, and Kris M Kitani. Going deeper into first-person activity recognition. In *Proc. CVPR*, 2016.

[61] Sibo Song, Ngai-Man Cheung, Vijay Chandrasekhar, Bappaditya Mandal, and Jie Liri. Egocentric activity recognition with multimodal fisher vector. In *ICASSP*, 2016.

[62] Yair Poleg, Ariel Ephrat, Shmuel Peleg, and Chetan Arora. Compact cnn for indexing egocentric videos. In *WACV*, 2016.

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

[64] Sanath Narayan, Mohan S Kankanhalli, and Kalpathi R Ramakrishnan. Action and interaction recognition in first-person videos. In *CVPR Workshops*, 2014.

[65] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013.

[66] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Proc. ECCV*, 2006.

[67] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the fisher kernel for large-scale image classification. In *Proc. ECCV*, 2010.

[68] Michael Wray, Davide Moltisanti, Walterio Mayol-Cuevas, and Dima Damen. SEMBED: Semantic Embedding of Egocentric Action Videos. In *Proc. ECCVW*, 2016.

[69] Dima Damen, Teesid Leelasawassuk, Osian Haines, Andrew Calway, and Walterio Mayol-Cuevas. You-Do, I-Learn: Discovering Task Relevant Objects and their Modes of Interaction from Multi-User Egocentric Video. In *Proc. BMVC*, 2014.

[70] Sven Bambach, Stefan Lee, David J. Crandall, and Chen Yu. Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions. In *Proc. ICCV*, 2015.

[71] Ilaria Gori, Jivko Sinapov, Priyanka Khante, Peter Stone, and JK Aggarwal. Robot-centric activity recognition in the wild. In *International Conference on Social Robotics*, 2015.

[72] Lu Xia, Ilaria Gori, Jake K Aggarwal, and Michael S Ryoo. Robot-centric activity recognition from first-person rgb-d videos. In *Proc. WACV*, 2015.

[73] Ilaria Gori, JK Aggarwal, Larry Matthies, and Michael S Ryoo. Multitype activity recognition in robot-centric scenarios. *IEEE Robotics and Automation Letters*, 1(1):593–600, 2016.

[74] Lu Xia and JK Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proc. CVPR*, 2013.

[75] Lin Sun, Kui Jia, Dit-Yan Yeung, and Bertram E Shi. Human action recognition using factorized spatio-temporal convolutional networks. In *Proc. ICCV*, 2015.

[76] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *Proc. ECCV*, 2016.

[77] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[78] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and ¡ 0.5 MB Model Size. *arXiv:1602.07360*, 2016.

[79] Michael S Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proc. ICCV*, 2011.

[80] Girmaw Abebe, Andrea Cavallaro, and Xavier Parra. Robust multi-dimensional motion features for first-person vision activity recognition. *Computer Vision and Image Understanding*, 149:229–248, 2016.

[81] Fatih Ozkan, Mehmet Ali Arabaci, Elif Surer, and Alptekin Temizel. Boosted Multiple Kernel Learning for First-Person Activity Recognition. *arXiv preprint arXiv:1702.06799*, 2017.

[82] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proc. ICCV*, 2013.

[83] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008.

[84] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proc. CVPR*, 2013.

[85] Lu Xia, Chia-Chih Chen, and JK Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Proc. CVPRW*, 2012.

[86] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning Deep Features for Discriminative Localization. In *Proc. CVPR*, 2016.

[87] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *Proc. CVPR*, 2011.

[88] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *Proc. ECCV*, 2016.

[89] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, 2017.

[90] Yansong Tang, Yi Tian, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Action Recognition in RGB-D Egocentric Videos. In *Proc. IEEE International Conference on Image Processing*, 2017.

[91] Christopher Zach, Thomas Pock, and Horst Bischof. A Duality Based Approach for Realtime TV-L 1 Optical Flow. In *Proc. Joint Pattern Recognition Symposium*, 2007.

[92] Alireza Fathi, Yin Li, and James M Rehg. Learning to recognize daily actions using gaze. In *Proc. ECCV*, 2012.

[93] Yin Li, Miao Liu, and James M Rehg. In the Eye of Beholder: Joint Learning of Gaze and Actions in First Person Video. In *Proc. ECCV*, 2018.

[94] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[95] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proc. CVPR*, 2015.

[96] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *Proc. NIPS*, 2016.

[97] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. ICCV*, 2015.

[98] Shikhar Sharma, Ryan Kiros, and Ruslan Salakhutdinov. Action recognition using visual attention. *NIPS workshop on Time Series*, 2015.

[99] Swathikiran Sudhakaran and Oswald Lanz. Convolutional Long Short-Term Memory Networks for Recognizing First Person Interactions. In *Proc. ICCV Workshops*, 2017.

[100] Swathikiran Sudhakaran and Oswald Lanz. Learning to Detect Violent Videos using Convolutional Long Short-Term Memory. In *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017.

[101] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. ActionVLAD: Learning Spatio-Temporal Aggregation for Action Classification. In *Proc. CVPR*, 2017.

[102] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *Proc. NIPS*, pages 34–45, 2017.

[103] Swathikiran Sudhakaran and Oswald Lanz. Attention is All We Need: Nailing Down Object-centric Attention for Egocentric Activity Recognition. In *Proc. BMVC*, 2018.

[104] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. Videolstm convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018.

[105] Pengfei Zhang, Jianru Xue, Cuiling Lan, Wenjun Zeng, Zhanning Gao, and Nanning Zheng. Adding Attentiveness to the Neurons in Recurrent Neural Networks. In *Proc. ECCV*, 2018.

[106] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. LSTA: Long Short-Term Attention for Egocentric Action Recognition. *arXiv preprint arXiv:1811.10698*, 2018.

[107] Wenbin Du, Yali Wang, and Yu Qiao. Recurrent spatial-temporal attention network for action recognition in videos. *IEEE Transactions on Image Processing*, 27(3):1347–1360, 2018.

[108] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proc. CVPR*, 2010.

[109] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proc. CVPR*, 2016.

[110] Tak-Eun Kim and Myoung Ho Kim. Improving the search accuracy of the VLAD through weighted aggregation of local descriptors. *Journal of Visual Communication and Image Representation*, 31:237–252, 2015.

[111] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proc. NIPS Workshop on Deep Learning*, 2014.

[112] Relja Arandjelovic and Andrew Zisserman. All about VLAD. In *Proc. CVPR*, 2013.

[113] Hilde Kuehne, Hueihan Jhuang, Rainer Stiefelhagen, and Thomas Serre. HMDB51: A large video database for human motion recognition. In *High Performance Computing in Science and Engineering 12*, pages 571–582. 2013.

[114] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[115] Liang Zhang, Guangming Zhu, Peiyi Shen, Juan Song, Syed Afaq Shah, and Mohammed Bennamoun. Learning spatiotemporal features using 3dcnn and convolutional lstm for gesture recognition. In *Proc. ICCVW*, 2017.

[116] Jason PC Chiu and Eric Nichols. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370, 2016.

[117] Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix Backpropagation for Deep Networks with Structured Layers. In *Proc. CVPR*, 2015.

[118] Yang Zhou, Bingbing Ni, Richang Hong, Xiaokang Yang, and Qi Tian. Cascaded interactional targeting network for egocentric video analysis. In *Proc. CVPR*, 2016.

[119] Hasan FM Zaki, Faisal Shafait, and Ajmal Mian. Modeling Sub-Event Dynamics in First-Person Action Recognition. In *Proc. CVPR*, 2017.

[120] Congqi Cao, Yifan Zhang, Yi Wu, Hanqing Lu, and Jian Cheng. Egocentric gesture recognition using recurrent 3d convolutional neural networks with spatiotemporal transformer modules. In *Proc. ICCV*, 2017.

[121] Sagar Verma, Pravin Nagar, Divam Gupta, and Chetan Arora. Making Third Person Techniques Recognize First-Person Actions in Egocentric Videos. In *Proc. ICIP*, 2018.

[122] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and Observer: Joint Modeling of First and Third-Person Videos. In *Proc. CVPR*, 2018.

[123] Yansong Tang, Zian Wang, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Multi-stream Deep Neural Networks for RGB-D Egocentric Action Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[124] Yang Shen, Bingbing Ni, Zefan Li, and Ning Zhuang. Egocentric Activity Prediction via Event Modulated Attention. In *Proc. ECCV*, 2018.

[125] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proc. CVPR*, 2018.

[126] Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan AlRegib, and Hans Peter Graf. Attend and Interact: Higher-Order Object Interactions for Video Understanding. In *Proc. CVPR*, 2018.

[127] Jingwen Wang, Wenhao Jiang, Lin Ma, Wei Liu, and Yong Xu. Bidirectional Attentive Fusion with Context Gating for Dense Video Captioning. In *Proc. CVPR*, 2018.

[128] Duy-Kien Nguyen and Takayuki Okatani. Improved Fusion of Visual and Language Representations by Dense Symmetric Co-Attention for Visual Question Answering. In *Proc. CVPR*, 2018.

[129] Junwei Liang, Lu Jiang, Liangliang Cao, Li-Jia Li, and Alexander G Hauptmann. Focal Visual-Text Attention for Visual Question Answering. In *Proc. CVPR*, 2018.

[130] Swathikiran Sudhakaran and Oswald Lanz. Top-down Attention Recurrent VLAD Encoding for Action Recognition in Videos. In *Proc. 17th International Conference of the Italian Association for Artificial Intelligence*, 2018.

[131] AJ Piergiovanni, Chenyou Fan, and Michael S Ryoo. Learning latent sub-events in activity videos using temporal attention filters. In *Proc. AAAI Conference on Artificial Intelligence*, 2017.

[132] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proc. CVPR*, 2016.

[133] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling Egocentric Vision: The EPIC-KITCHENS Dataset. In *Proc. ECCV*, 2018.

[134] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The something something video database for learning and evaluating visual common sense. In *Proc. ICCV*, 2017.

[135] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Proc. ECCV*, 2016.

[136] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfruend, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. *arXiv preprint arXiv:1801.03150*, 2018.