A Dissertation Report

on

Action Recognition using Feature Fusion

Submitted in the fulfillment of the requirements

for the award of degree

Master of Technology

in

Computer Science and Engineering

Submitted By:

RENUKA SHARMA

Enrollment Number: 16535035

Under the supervision of

DR. BIPLAB BANERJEE

Assistant Professor, Dept. of Computer Science and Engineering

Department of Computer Science and Engineering

Indian Institute of Technology Roorkee

May, 2018

Candidate's Declaration

I hereby declare that the work presented in this dissertation "**Action Recognition using Feature Fusion**" towards fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period of **May 2017 to May 2018** under the supervision of **Dr. Biplab Banerjee**, Assistant Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, India.

The content presented in this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: .....
Place: Roorkee

(Renuka Sharma)

Certificate

This is to certify that Thesis Report entitled "**Action Recognition using Feature Fusion**" which is submitted by Renuka Sharma (16535035), towards the fulfillment of the requirements for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science & Engineering, Indian Institute of Technology Roorkee, India** is carried out by her under my esteemed supervision and the statement made by the candidate in declaration is correct to the best of my knowledge and belief.

Date: .....
Place: Roorkee

(Dr. Biplab Banerjee)
Assistant Professor
Department of Computer Science & Engineering
Indian Institute of Technology Roorkee

Acknowledgements

I would have never been able to complete my dissertation without the constant support and guidance of my supervisor, my friends and family.

Firstly, I would like to express my sincere gratitude to my supervisor Dr. Biplab Banerjee for being there and guiding me in every complication I faced during my research work and supporting me in every way possible in M.Tech. study and related research, for his patience, motivation, immense knowledge and the most helping attitude. His guidance helped me in all the time of research and writing of this thesis.

I would like to thank my lab (Tinkering Lab) coordinators for providing me the needed configuration of the system and helping with the issues I faced while working there.

I am grateful to the Department of Computer Science and Engineering of IIT Roorkee for providing valuable resources to aid my research. A hearty thank to my friends and family for encouraging me all the while.

RENUKA SHARMA

Abstract

Action recognition seems to be a very easy task for us humans but it requires a lot of information processing in terms of recognizing patterns when it comes to computer systems. Here, we try to devise a new way of action recognition for intelligent systems by fusing the shallow and the deep features from the data. Shallow feature extraction starts by identifying the motion salient pixels first, thus eliminating unwanted information and then extract the improved trajectory information from it. To get the deep features, we make use of Convolutional Neural Network (CNN). There will be separate classifiers for both the deep features and shallow features which will be fused in order to result in an efficient classifier for the action recognition. We are using HMDB-51[1] video dataset, one of the most challenging datasets for action recognition which consists of various actions of different kinds like clap, run, walk, box, etc taken from various sources like YouTube, movies and Google videos under various illumination effects, occlusion, camera angle variation and pose variation.

Keywords: Action recognition, motion saliency, feature extraction, feature encoding, dense trajectory, improved trajectory, classifier fusion

Contents

List of Figures

List of Tables

Abbreviations

BOP	Bag of 3D points
CCTV	Closed Circuit Television
CNN	Convolutional Neural Network
GBVS	Graph Based Visual Saliency
HOF	Histogram of Optical flow
HOG	Histogram of oriented Gradients
MBH	Motion Boundary Histograms
VOE	Video Object Extraction

Chapter 1

Introduction

The human action is a sequence of body movements which varies from the simplest movement of a limb to complex joint movement of a group of limbs and body.

Action recognition is the task of identifying the activities going on in the videos. The human activities sometimes involve other objects too and these activities vary in spatial and temporal domain. Human beings can easily distinguish such actions due to the availability of the visual cortex in the brain. But, for the computer systems to do the same, expert designed systems would not suffice and the deep learning systems need to be trained on the data so as to learn the discriminative features.

In the field of Computer Vision, Human action recognition has been an active topic. This is due to the need to automatic video analysis like in visual surveillance, sports video analysis, video retrieval and human machine interfaces. Based on the complexity, human activities are divided into the following categories: gestures, actions, interactions and group activities. The basic process of action recognition covers identification of the human and its body part, tracking the identified parts, and then associating a label with the tracking of parts you have done.

In order to identify the action of the human being, intelligent systems will need to track the movement of body parts in the action video and then match it with the previously defined patterns and then assign it a label with the help of the apt classifier it is using in order to distinguish the patterns and helps in assigning the apt labels.

1.1 *Motivation*

Human action recognition is an important area in the field of computer vision due to increasing security and day to day concerns. Nearly all the shops, malls, stations and institutions have Closed Circuit Television (CCTV)

FIGURE 1.1: Different human actions in videos for recognition (link)

cameras installed so that human activities could be tracked manually and the suspicious activities could be identified as a precautionary measure.

Nowadays the technology is advancing so much that one day, in the near future, all the security systems will be tracked by the intelligent systems and the work is going on in all the fields to enhance the capabilities of the intelligent systems that will be able to assist us and ease our work. So, human action recognition is also an important area of research and here we are finding some more efficient ideas than the existing ones to get there. The reason being, we need intelligent systems to observe the videos for visual surveillance, human-machine interfaces, sports video analysis and video retrieval in the best way possible.

1.2 *Organization of the Report*

The report is categorized in parts. The *Literature Review* part will cover the major work done in the action recognition area which has inspired us and lays the foundation of our work. We will list the gaps that all the previous methods had and if they can be improved by our method.

Next chapter, *Problem definition* covers the formulation of the problem that we are dealing with by enlisting the major subproblems to be solved in order to solve the problem at hand.

In the *Proposed solution* chapter, we will propose our approach to solve the problem enlisting all the steps we are performing for it.

The *Implementation* chapter will consist of the assumptions and inputs, datasets taken in our case for solving the problem and will cover the results for various approaches we performed.

In the *Conclusion* chapter, we discuss about the results we have got for the implementation of our proposed solution.

At the end *Future work* chapter consists of all the possible improvements that can be done in the future.

Chapter 2

Literature Review

Single layered approaches and hierarchical approaches have been implemented by various authors for the action recognition in the video data. [2] [3] [4]

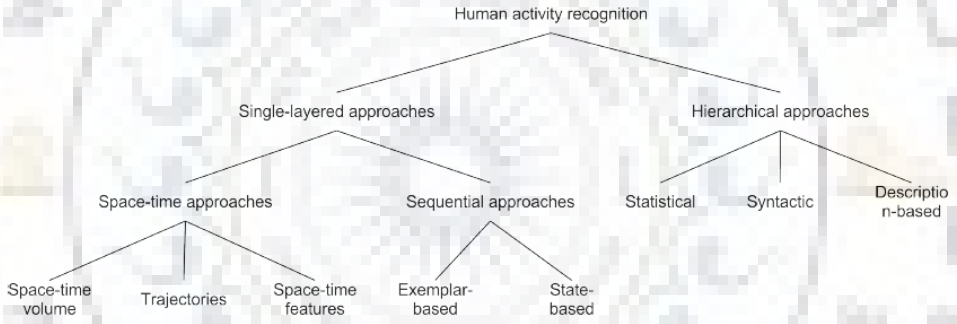

FIGURE 2.1: Different methods for action recognition [4]

In [4], the human action recognition task has been defined to be implemented in two major ways viz. Single layered approach and Hierarchical approach which is further divided into sub-approaches.

2.1 *Single layered approaches*

The single layered approach the activities are recognized from the raw video data rather than sub-actions or sub-activities. Thus simple video datasets like KTH are used with this method. Single layered approaches are used to identify simple actions and it can thus be used to assist in more complex action recognition tasks such as hierarchical action recognition based approach.

The single layered approaches for action recognition have been categorized into space-time approaches and sequential approaches. The key distinguishing feature among the two is how the third (tempral) dimension is treated.

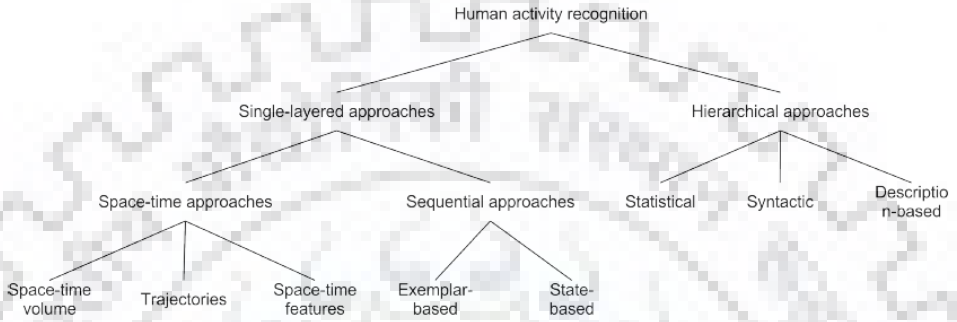

FIGURE 2.2: Taxonomy of human activity recognition methods [4]

2.1.1 *Space Time approaches*

In space time approaches, the time is considered as a third, spatial dimension and videos are extracted as 3D volume at each time instant which can be compared with another 3D volume for feature extraction. Considering the space time volume contains the information which can identify the motion and activities, various algorithms have been introduced to identify the motion patterns in the space time volume.

It has further been categorized into:

- Action recognition with space time volume [5]
- Action recognition with space time Trajectories
- Action recognition with Space time features [6]

Our focus is on action recognition using space-time trajectory based approaches which are based on the concept that tracking of joint positions is sufficient to recognize human actions.[7]

In [8] the focus is on recognizing human actions using body joints which are extracted from sequences of depth maps.

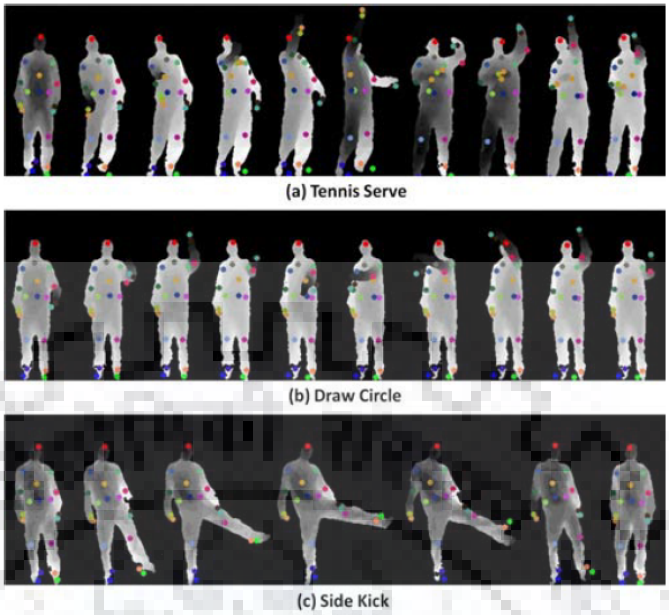

FIGURE 2.3: Sequences of depth maps for body joints [8]

2.1.2 *Sequential approaches*

These approaches consider human movement as ordered observation in time. These take sequential relationships into consideration, that's why these achieve better results than their space-time counterpart methods. There are various types of sequential approaches:

- Examplar based approach [9] [10]
- State based approach [11]

The focus of exemplar based approach is how a video can be compared to a smaller template video in order to identify the action sequence. In the State based model, it learns the state model and thus map each video as a set of states that will eventually be identified with the help of a classifier.

2.2 *Hierarchal approaches*

In [12], it describes hierarchal approaches make use of simpler or low-level sub-activities to recognize interesting events(high level activities). Hierarchical approaches have a close relationship with single-layered approaches to some extent.

Using the taxonomy proposed in[12], these are categorized into 3 groups:

- statistical approaches
- syntactic approaches
- description-based approaches

By study it has been found that hierarchical representation is better than non- hierarchical bag- of -words representation.

In the single layered approach, just the frames extracted from video after sampling are scanned in order to identify the action but in the case of hierarchical approach, the data set is scanned again and again to get the features from the sub-section created to extract the features and eventually after a few passes, according to the need of the method, we are able to identify the action of the human.

In [13], the recognition of human actions has been done from the sequence of depth maps. Effective and efficient use of depth information is required to design algorithms for action recognition using depth map sequences. In this paper, efforts have been made to devise a method for action recognition that does not require joint tracking. A bag of 3D point (Bag of 3D points (BOP)) is to be extracted from depth map sequence to get the action performed by the person. A training sample is to be created for mapping of a posture to the particular action. Every action has been encoded in the form of action graph which can have one or more paths and each node in such a graph defines a posture which is salient for that particular action. Experiments have shown that in order to achieve 90% of action recognition accuracy, only 1% of sampling is enough from 3D point depth maps.

The [14] paper titled "Graph Based Visual Saliency (GBVS)" is the base paper used for the identification of the salient pixels in the given video so as to track them in the consecutive frames.

In [15] and [16], optical flow has been used to find the trajectory of motion in the video frames unlike the above works where depth maps are used for which we had to take into account depth map for each frame of video which would require the use of extra hardware. As dense sampling has been very much successful in image classification, an approach to describe videos by dense trajectories has been proposed. In each consecutive frame of video dense optical flow displacement has been captured after identification of dense sampling of video. A descriptor based on Motion Boundary Histograms (MBH) (Motion Boundary Histograms) has been introduced. Bag-of-features approach has been used for action classification.

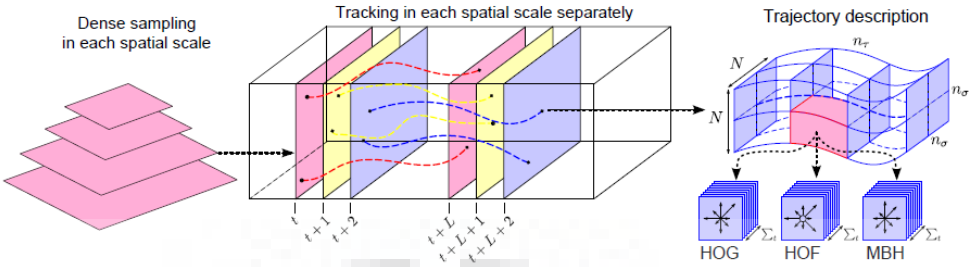

FIGURE 2.4: Dense trajectory description [16]

In [15], Wang et al. are making use of Histogram of oriented Gradients (HOG), Histogram of Optical flow (HOF) and MBH as the features extracted for making the classifier for human action recognition. We will make use of [17] for separating foreground and background regions in and across video frames. It makes use of motion saliency feature for this.It basically does Video Object Extraction (VOE).

FIGURE 2.5: Dense trajectory depiction (link)

The red dots define the pixels not in motion and green ones are moving ones. The green pixels will be focused upon for deep feature extraction purpose. Then after identification of the deep features, these features need to be associated with a set of actions and at the end every action will have a path of postures the person goes through that will be the key postures used for identification of the action. We design an algorithm to train our classifier which will do action recognition this for us just by giving an input video of a person.

The paper [18] titled "Action Recognition with Improved Trajectories" can be considered the base paper for us as the improved trajectory method mentioned in this paper for the tracking of the tracking of feature points in

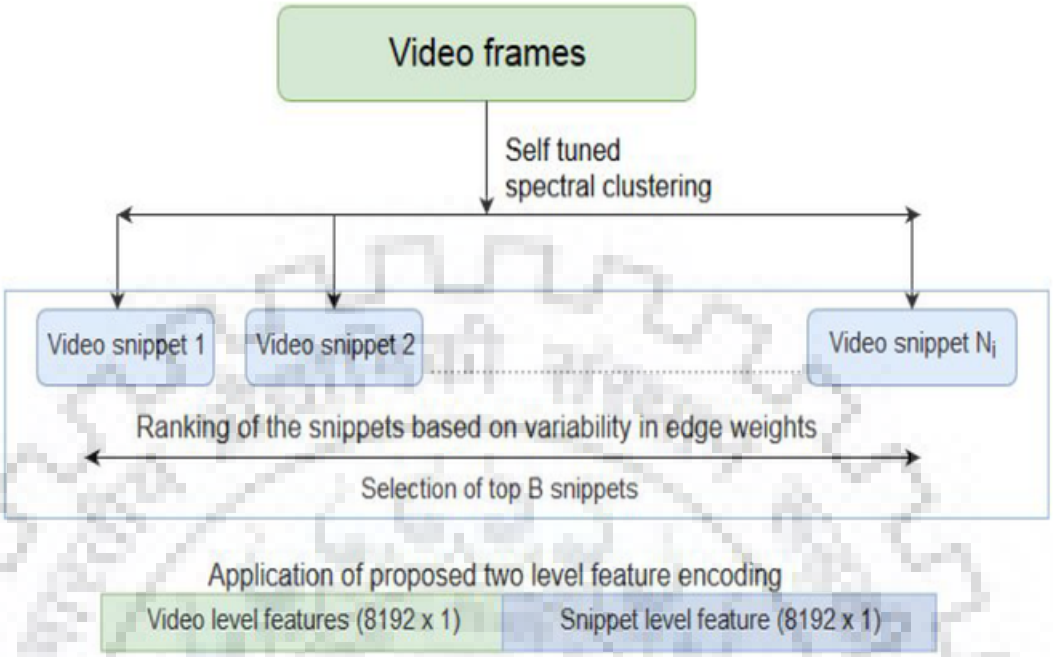

FIGURE 2.6: Proposed feature encoding scheme for [20]

the spatio tempral domain is the state of the art method and it is the one we will be using for feature tracking.

In [19], feature encoding techniques have been described which we will be taking as a reference for our method.

[20] paper has been referenced for the purpose of deep feature extraction from the trajectory obtained by the improved trajectory approach. It divides the video into snippets that may be overlapping in nature and thereafter find the video level and snippet level features in the input video which would give us the deep features (see Fig 2.3).

In [21], CNNs have been presented as a powerful tool for solving image recognition problems. Since, CNN require large amount of time to get trained, it is advised to divide it into 2 separate processing streams: one will be operating on low-resolution frames and the other will be operating on high-resolution middle frames which are much useful in an image. Thus the dimensionality is reduced but the classification accuracy remains the same.

There are various fusion schemes present in order to fuse two features.

The fusion can be at the feature level or the classifier level. In [22], the fusion of two deep networks is performed in order to classify the videos. Here three Convolutional Neural Networks are trained to model spatial, short-term motion and the audio clues respectively. This approach performs better than the state of the art on UCF-101[23] and Columbia Consumer Videos dataset.

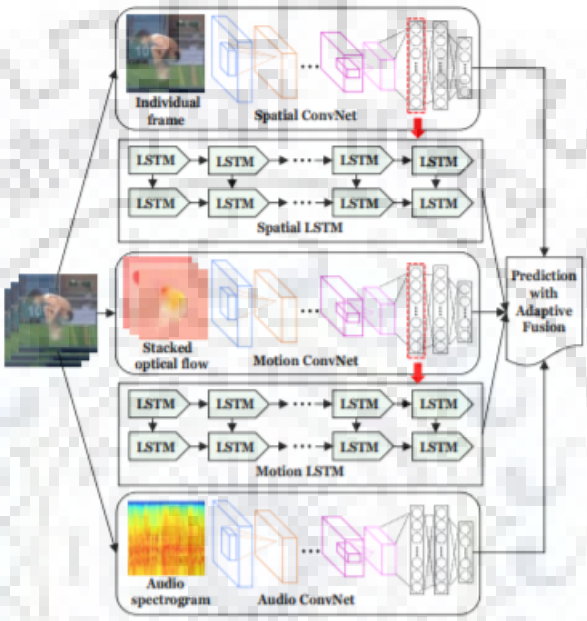

FIGURE 2.7: Proposed multi-stream video classification framework in [22]

Apart from feature level fusion, we have classifier level fusion where there could be n number of classifiers which are already trained on the same dataset and we are not sure about the individual results generated by them. Then, in order to solve this, the classifiers are combined together to form Multi Classifier System (MCS) which is a structured way to combine the outputs of the individual classifiers. MCS can be characterized by:

- Architecture
- Fixed/trained combination strategy

The MCS architecture could be serial, parallel or hybrid. Also, fusion is useful only when the combined classifiers are complementary to each other. That way, they all add up to result in the best of classifier possible.

2.3 *Gaps Identified*

The papers [8], [9], [13] work on special type of input videos and not just plain 2D videos which are readily available. The edge joint based approach for action recognition requires the videos to contain the depth maps too so as to get the positions of joints present in the human body.

The paper [15], titled "Action recognition using Dense Trajectories" was the first most efficient approach invented for the feature tacking but it lacks in dealing with the camera motion. In order to overcome that, [18] improved trajectories were introduced which omitted the trajectory formation due to camera motion.

Both [15] and [18] have done feature encoding just after getting the trajectories. We can do deep feature extraction using CNN before going for the feature encoding and that's what we are trying to do.

Chapter 3

Problem Definition

We are to get the input video, of a person performing some action and by the help of some human activity methods or a combination of them, we need to extract the features of person present in the video frames, find the relationship between them and eventually identify the action the person is performing by the help of mapping function (classifier) which has been trained already. The labels will be created beforehand corresponding to each sequence of postures by the dataset that we will be using like HMDB51, UCF101.

Like, in the case of jogging, a person acquires a lot of postures in between and we need to identify and associate those key postures and train our classifier in such a way to identify and map those features in the input video that it gets later to recognize action of the person.

The problem of ours can be broken down into 3 main subproblems:

- **Shallow feature extraction** After preprocessing step, which involves extraction of frames from the videos and resizing them according to the need, we will have to identify the motion salient pixels in each frame which will remain our point of concern in the future process of action recognition.

  Then in order to track the features in the frames, we will consider only those salient pixel regions as they are the ones covering the useful information regarding the human action at the local level. The improved trajectories will have to be generated for the motion salient regions only.

- **Deep Feature extraction in frame**
  Let X be a video input represented by a set of frames:

$$X = \{x_1, x_2, x_3, ..., x_n\}$$

where $x_i \in R^N$ is a feature of the frame i extracted.

These features identify the videos at the local level as well as the global level (video level).

- **Action recognition classifier fusion**
  We need to find a classifier h as follows:
  $h : X \rightarrow Y$ where Y is the action category to which input video X is mapped.

  This h is a fusion of two classifiers $h1$ and $h2$ which are Shallow feature based classifier and Deep feature based classifier each. The fusion scheme should be such that both the classifiers are equally taken into account.

Chapter 4

Proposed Solution

Researchers have adopted single layered approaches as well as hierarchical approaches for the identification of the human action. We are thinking of using single layered space time dense trajectory approach along with hierarchical approach for deep feature extraction.

- **Preprocessing of dataset** Firstly, the dataset needs to be extracted and placed in directories according to their specific labels in the video form. Another set of dataset directory will be made for the frames extracted from each of the videos which will be later worked up on for motion salient pixel detection and for deep features extraction.

- **Motion salient pixels extraction** The preprocessed video frames is then filtered to get the motion salient pixels in the frames using the method of GBVS [14]. The method basically consists of 3 steps, namely:

  - Extraction of feature vectors
  - Activation map formation
  - Normalization of Activation map

  In GBVS, a graph G_N is constructed with n^2 nodes. For each node (i,j) and every other node (p,q) with which (i,j) is connected is given weight:

  $$w((i,j),(p,q)) := A(p,q).F(i-p,j-q) \tag{4.1}$$

  where A(p,q) is the activation function and F(a,b) is the closeness of a and b. The weight is proportional to the dissimilarity and closeness in the domain.

  We thus get the saliency values for all the pixels in the image and we can choose whatever threshold suitable for us.

- **Shallow feature extraction from the frames** The frames thus obtained will be mapped to the consecutive frame in the video to identify the

FIGURE 4.1: Image given for GBVS saliency detection

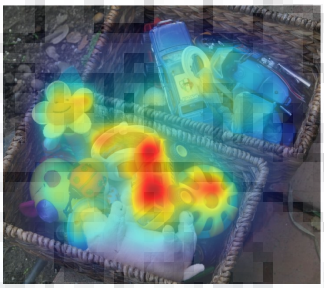

FIGURE 4.2: Output of GBVS saliency

trajectories of various pixels present in the salient region with the help of optical flow. For this purpose improved trajectory approach is the best one present right now and we have used that approach.

Improved trajectory feature extraction is the current state-of-the-art method present right now. It improves the dense trajectory approach by taking into account the camera motion in order to correct the camera angle changes and other illumination effects that are present in the video data taken from movies, YouTube which may involve shadow, occlusion, pose variation, illumination effects, etc.

- **Deep feature extraction corresponding to trajectories** Then, hierarchical approach will be applied to efficiently access the features of the video [20] by getting Video level features and snippet level features.

  We make use of VGGf-net Convolutional Network for deep feature extraction at each frame and then the gradient is to be calculated for each of the consecutive frames as follows:

  Considering n frames, the features are extracted as:

$$f = f_1, f_2, f_3, ..., f_n \tag{4.2}$$

where f_i are the frame level features for $i = 1$ to n

Then in order to compute the gradients: $f_{21} = f_2 - f_1$

.

. $f_{nn-1} = f_n - f_{n-1}$ Thus we store these positive and negative gradients in a vector which will then be fed to the classifier.

- **Fisher encoding for videos** It is performed on the improved trajectory shallow features extracted so as to train the classifier with them.

  In the fisher encoding[24], first the Gaussian Mixture Model (GMM) is to be prepared for the whole dataset which will give the mean, covariance and prior values for the whole dataset.

  $$[means, covariance, priors] = vl\_gmm[data, numClusters] \tag{4.3}$$

  We thus get the model for the whole dataset. In order to encode each of the video files, fisher vector encoding is to be used as follows:

  $$encoding = vl\_fisher(dataToBeEncoded, means, covariances, priors) \tag{4.4}$$

  In the *dataToBeEncoded* variable, we give the individual video's improved trajectory and after the above step, we will get the fisher encoding in the *encoding* variable.

- **Training classifier for shallow features** The Linear Support Vector Machine (SVM) classifier will then be designed for the shallow features encoded with the help of fisher encoding and we will get the probabilistic analysis of class to which the features will belong to.

- **Training classifier for deep features** The Linear SVM will also be applied on the gradient vector for the deep features extracted which has the positive and negative gradients. The SVM will be trained on the basis of these gradient values in the video data.

- **Classifier fusion** In order to fuse the classifiers, we are giving both the classifiers equal weightage by averaging the probability values for both deep classifier and shallow classifier probabilities p_1 and p_2 respectively. We will get $p = (p1 + p2)/2$
  where p is the probability with which the data will belong to the particular class. So, we get the classifier mapping function h which will map the video data X to the correct action label Y.

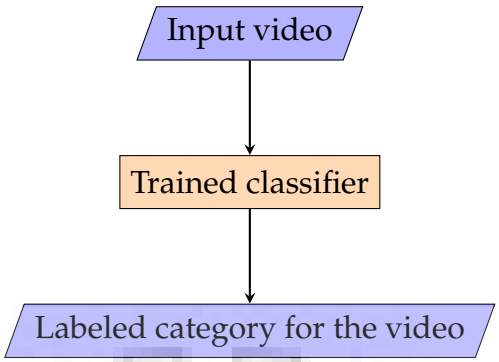

FIGURE 4.3: Usage of the classifier

- **Apply classifier on new input** For every input video, we will apply this classifier to identify the action performed in the video.

In this approach, we will make use of 2D image sequence generated from the video which is easily available unlike the case of depth maps, since we are using motion saliency for salient pixel detection in the video frames.

Motion saliency is to be used to identify salient pixels in the video frames and improved trajectory with optical flow will identify the pixels in motion and then we will do deep feature extraction, a new method to be devised for that. This way a classifier will be designed and trained appropriately which will be put to use thereafter.

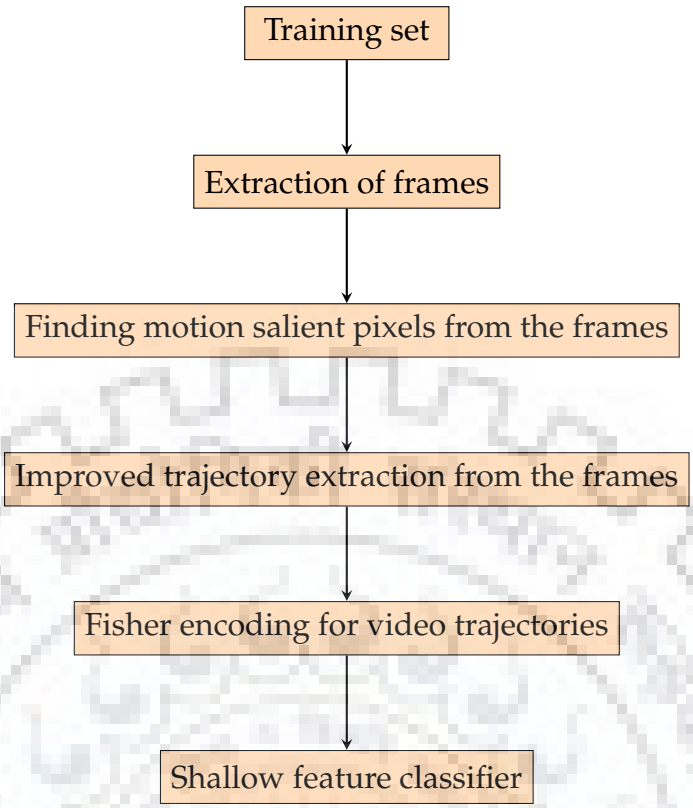

FIGURE 4.4: Classification on the basis of shallow features

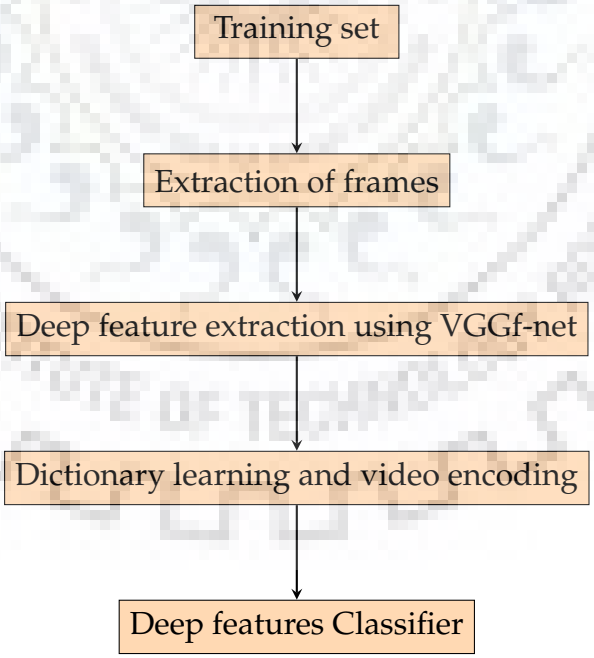

FIGURE 4.5: Classification on the basis of deep features

Chapter 5

Implementation

5.1 *Datasets Used*

We have used HMDB-51 [1] dataset which is the large video database covering 51 categories of human actions in different backgrounds. It has 6,766 video clips taken from Digital movies to You Tube and those had been manually labeled, containing actions like brush hair, pick, pull, cartwheel, run, ride horse, etc.

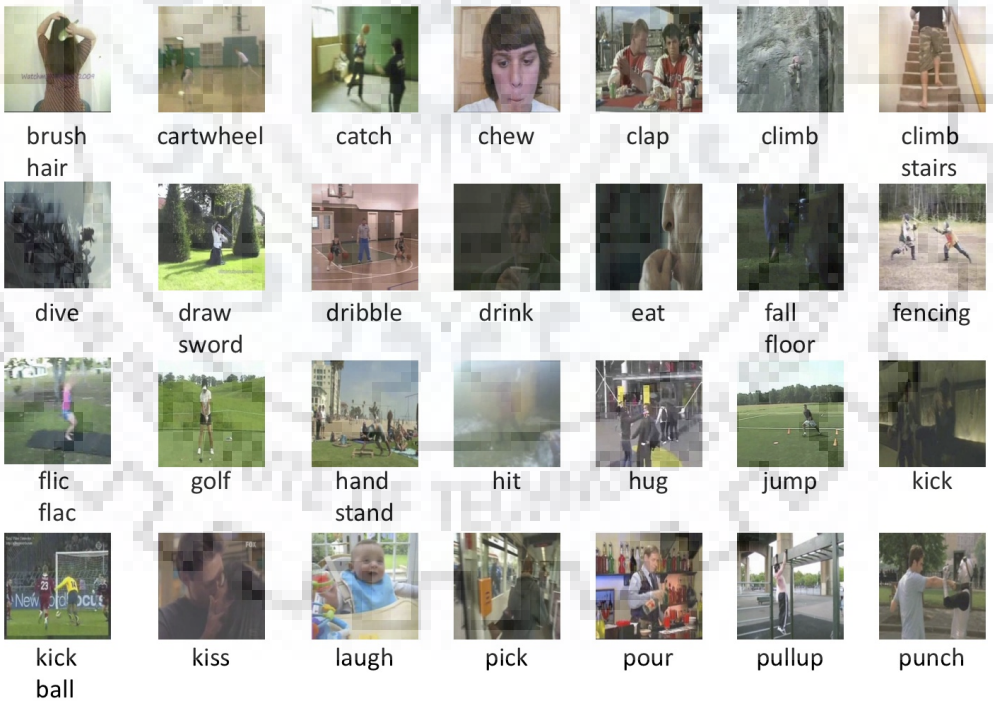

FIGURE 5.1: HMDB51 action classes

We will be using the UCF-50 [25] dataset too for our experimental purpose which contains 50 action categories consisting of the videos taken from

You Tube. It is in continuation to the UCF-11 dataset consisting of 11 action categories.

5.2 *Experiments*

5.2.1 *Shallow features extraction*

We have started with the HMDB-51 dataset for the experiments as it contains diverse set of action categories and it is quite challenging too. The dataset is first sampled into frames which are then resized to 150X150 size so that the GBVS visual saliency could be applied smoothly.The result of the GBVS code is the file containing the saliency values for all the pixels in the frame. The salient pixels are the pixel values greater than a threshold value of 0.20 which will be meaningful to us. Those pixels are assigned a value 1 and the rest 0 to form the following type of frames. These are the pixels in the frame that we should be concerned about.

(A) Salient pixels

(B) for this frame

FIGURE 5.2: Motion salient pixels detection

The following figure contains for the trajectory of the frames using improved trajectory approach.

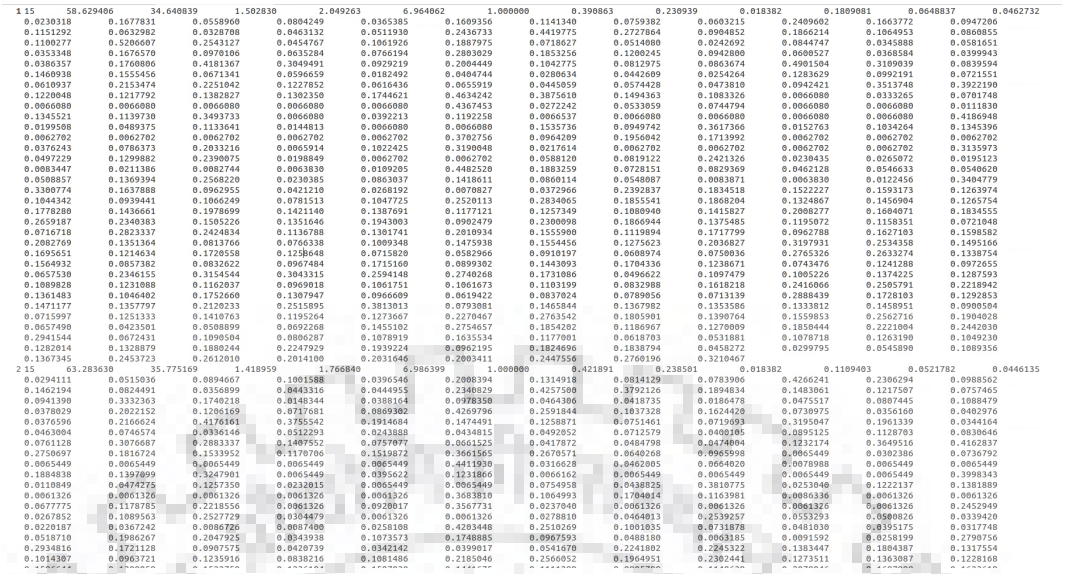

FIGURE 5.3: Improved trajectory for a video

The improved trajectories are for each of the frames and it contains the mean_x, mean_y, HOG, HOF and MBH descriptors. Thus we are able to perform the feature fusion for our approach.

The training is done using Linear SVM's multiclass [26] approach.

5.2.2 *Fisher Encoding*

The trajectory features thus obtained from improved trajectory approach needs to be encoded in order to form a feature vector which will be used to represent the data eventually. We are making use of the VLFeat [27] open source library in MATLAB which is used for image understanding, local features extraction and matching. It contains algorithms like VLAD, Fisher, SIFT.

In order to encode the features we first have to get the GMM(Gaussian Mixture Model) foe the whole data in order to get the mean, covariance and priors for the whole dataset. This has to be run for all the video files present in the dataset.

$$[means, covariance, priors] = vl\_gmm[data, numClusters] \tag{5.1}$$

Here, data is the trajectory features for all the files and it is a matrix of dimension 496 X 6766 where 496 are the dimensions (HOG+HOF+MBH) and 6766 is the number of data (video files) present in the HMDB51 dataset, and we are taking 64 to be the numClusters, number of clusters for the data.

We thus get the model for the whole dataset. In order to encode each of the video files, fisher vector encoding [24] is to be used.

$$encoding = vl\_fisher(dataToBeEncoded, means, covariances, priors) \quad (5.2)$$

In the above equation, the *means*, *covariances*, and *priors* are taken from the equation "5.1" above. This has to be run for each of the video's trajectories separately in order to get the encoding corresponding to them.

5.2.3 *Deep features extraction*

FIGURE 5.4: VGGf CNN's layer 1 parameters

The base paper used for deep feature extraction is [20]. It extracts video level features and snippet level features from the same video data and then compute the gradients for them.

The data to be fed for classification is a set of positive and negative gradients over the video and it will represent the video itself.

The Linear SVM is then used to classify this data.

5.2.4 *Classification using Linear SVM(Support Vector Machines)*

We are doing the classification with the help of multi class Linear SVM using LIBSVM library in MATLAB. It gives the probability with which it thinks the data belongs to a particular label.

The Linear SVM is applied on the deep features extracted to get the Deep feature classifier. Linear SVM is applied on shallow features as well to get the Shallow feature classifier.

5.2.5 *Classifier Fusion*

The Deep Feature classifier and Shallow Feature Classifier will have probability values p1 and p2 for Deep and Shallow classifier each.

$$p = (p1 + p2)/2 \tag{5.3}$$

We perform the average of the two probabilities *p1* and *p2* to get the average probability *p* which will give the idea about the probability with which the video data belongs to a particular class label.

TABLE 5.1: Results for HMDB 51 dataset

Descriptors Used	Baseline	Shallow features	Deep Features	Shallow + Deep
Trajectory	25.4%	42.1%	60.8%	63.4%
HOG	38.4%	35%	45.2%	41%
HOF	39.5%	50%	50.5%	50.1%
MBH	49.1%	60%	46.6%	61.1%
HOF+MBH	49.8%	55%	52.4%	59.8%
Combined	52.2%	55%	58%	60.5%

Chapter 6

Conclusion

Classifier fusion is obviously useful if the combine classifiers are mutually complementary. Clearly, The shallow features learning classifier and the deep features learning classifier are complementary to each other in the way one is extracting the features on the shallow level thus keeping equivalent focus on the entire data at hand whereas in the Deep feature approach, the features and extracted on both the local level and the whole video level both.

As we are taking the average of the probabilities of both the classifiers, so both the classifiers are equally considered in order to reach the decision of classifying the correct action label to the video data given at hand.

The accuracy of the cation recognition model has clearly increased by the combination of the deep level features and shallow level features.

If there would have been more than 2 classifiers for this purpose, we can use MCS (Multiple Classifier System) which can be considered as Multiple Expert System or Committee of Experts or Mixture of Experts or Classifier Ensemble or Composite Classifier System. The efficiency of such system depends on the architecture that we are using for the combination of such classifiers. It could be serial, parallel or hybrid.

Chapter 7

Future Work

As we are using motion saliency so there is no need to take a particular type of video as input, needed for depth map based methods, any 2D video will suffice as input in our case and there are variety of such datasets available for experiments.

Our method is the efficient one in terms of computation done in order to detect the trajectory for the video frames as the first step we are performing is to identify the salient pixels from each of the frames and considering only those trajectories falling under the salient area and getting the improved trajectory for it resulting in the shallow features. For the purpose of motion salient pixels detection, I am using GBVS which could have been replaced with better saliency approach in accordance with the challenging dataset at hand.

For future work, we will be taking larger datasets, like UCF101 [23] which consists of 27 hours of video data, covering 101 action categories and has around 13k video clips. It is the most challenging dataset present right now as it has the videos with large variations in illumination conditions, camera motion, background clutter, viewpoint change, scaling, object appearance and pose.

And also, as we have taken two classifiers, one is extracting the deep features and the other shallow features and then we are fusing them. We can also use a third classifier which may take into account the temporal order of the video frames.

Bibliography

[1] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "Hmdb: A large video database for human motion recognition", in *2011 International Conference on Computer Vision*, 2011, pp. 2556–2563. DOI: 10.1109/ICCV.2011.6126543.

[2] S. Herath, M. Harandi, and F. Porikli, "Going deeper into action recognition: A survey", *Image and Vision Computing*, vol. 60, pp. 4–21, 2017.

[3] Z. Shu, K. Yun, and D. Samaras, "Action detection with improved dense trajectories and sliding window", in *Workshop at the European Conference on Computer Vision*, Springer, 2014, pp. 541–551.

[4] G. Cheng, Y. Wan, A. N. Saudagar, K. Namuduri, and B. P. Buckles, "Advances in human action recognition: A survey", *ArXiv preprint arXiv:1501.05964*, 2015.

[5] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates", *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[6] I. Laptev and T. Lindeberg, "Velocity adaptation of space-time interest points", in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, IEEE, vol. 1, 2004, pp. 52–56.

[7] G. Johansson, "Visual motion perception.", *Scientific American*, 1975.

[8] X. Yang and Y. L. Tian, "Eigenjoints-based action recognition using naive-bayes-nearest-neighbor", in *Computer vision and pattern recognition workshops (CVPRW), 2012 IEEE computer society conference on*, IEEE, 2012, pp. 14–19.

[9] D. M. Gavrila and L. S. Davis, "Towards 3-d model-based tracking and recognition of human movement: A multi-view approach", in *In International Workshop on Automatic Face- and Gesture-Recognition. IEEE Computer Society*, 1995, pp. 272–277.

[10] T. Darrell and A. Pentland, "Space-time gestures", in *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on*, IEEE, 1993, pp. 335–340.

[11] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model", in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, IEEE, 1992, pp. 379–385.

[12] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review", *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[13] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points", in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, IEEE, 2010, pp. 9–14.

[14] J. Harel, C. Koch, and P. Perona, "Graph-based visual saliency", in *NIPS*, 2006.

[15] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories", in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, IEEE, 2011, pp. 3169–3176.

[16] ——, "Dense trajectories and motion boundary descriptors for action recognition", *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013.

[17] W.-T. Li, H.-S. Chang, K.-C. Lien, H.-T. Chang, and Y.-C. F. Wang, "Exploring visual and motion saliency for automatic video object extraction", *IEEE transactions on image processing*, vol. 22, no. 7, pp. 2600–2610, 2013.

[18] H. Wang and C. Schmid, "Action recognition with improved trajectories", in *The IEEE International Conference on Computer Vision (ICCV)*, 2013.

[19] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: An evaluation of recent feature encoding methods.", in *BMVC*, vol. 2, 2011, p. 8.

[20] B. Banerjee and V. Murino, "Efficient pooling of image based cnn features for action recognition in videos", in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2637–2641. DOI: 10.1109/ICASSP.2017.7952634.

[21] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks", in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[22] Z. Wu, Y. Jiang, X. Wang, H. Ye, X. Xue, and J. Wang, "Fusing multi-stream deep networks for video classification", *CoRR*, vol. abs/1509.06086, 2015. arXiv: 1509.06086. [Online]. Available: http://arxiv.org/abs/1509.06086.

[23] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild", *CoRR*, vol. abs/1212.0402, 2012.

[24] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action Recognition with Stacked Fisher Vectors", pp. 581–595, 2014.

[25] K. K. Reddy and M. Shah, "Recognizing 50 human action categories of web videos", *Mach. Vision Appl.*, vol. 24, no. 5, pp. 971–981, Jul. 2013, ISSN: 0932-8092. DOI: 10.1007/s00138-012-0450-4. [Online]. Available: http://dx.doi.org/10.1007/s00138-012-0450-4.

[26] Cody, *Multi class svm*, http://www.mathworks.com/matlabcentral/fileexchange/33170-multi-class-support-vector-machine/, December 2012.

[27] A. Vedaldi and B. Fulkerson, "Vlfeat: An open and portable library of computer vision algorithms", in *Proceedings of the 18th ACM International Conference on Multimedia*, ser. MM '10, Firenze, Italy: ACM, 2010, pp. 1469–1472, ISBN: 978-1-60558-933-6. DOI: 10.1145/1873951.1874249. [Online]. Available: http://doi.acm.org/10.1145/1873951.1874249.