Human Action Recognition in Videos Using Intermediate Matching Kernel

Sharath chada

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Department of Computer Science Engineering

July 2014

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

(Signature)

(Sharath chada)

CS12M1001

(Roll No.)

Approval Sheet

This thesis entitled Human Action Recognition in Videos Using HMM based Intermediate Matching Kernel by Chada Sharath is approved for the degree of Master of Technology from IIT Hyderabad.

Dr. Sumohana Channappayya(Examiner)
Department of Electrical Engineering,
IIT Hyderabad

N.B. Vineeth

Dr. Vineeth N Balasubramanian(Examiner)
Department of Computer Science and Engineering,
IIT Hyderabad

C. Krishna Mohan

Dr. C KrishnaMohan(Adviser)
Department of Computer Science and Engineering,
IIT Hyderabad

Dr. Subramanyam Kalyanasundaram(Chairman)
Department of Computer Science and Engineering,
IIT Hyderabad

Acknowledgements

Many individuals contributed in many different ways to the completion of this thesis. I am deeply grateful for their support, and thankful for the unique chances they offered me.

Im greatly thankful to my supervisor Dr. C Krishna Mohan for his valuable guidance, constant encouragement and timely suggestions. I would like to make a special mention of the excellent facility provided to me by IIT Hyderabad.

I'm also thankful to the support provided by Mr. Mettu Srinivas and Mr. Debaditya Roy.

More than to anyone else, I owe to the love and support of my family. My father Ramesh Chada, my mother Narmada Chada and my elder sister Keerthana Chada.

Abstract

Human action recognition can be considered as the process of labelling the videos with the corresponding action labels. Coming to the fields of computer vision, video sensing this has become an important area of research. There are a lot of factors such as recording environment,intra class and inter class variations,realistic action ambiguities and varying length of actions in the videos which make this problem more challenging

Videos containing human actions can be considered as the varying length patterns because the actions in videos may last for different duration. In this thesis the issue of varying length patterns is being addressed. To solve this issue a paradigm of building intermediate matching kernel as a dynamic is used so that the similarity among the patterns of varying length can be obtained. The idea of the intermediate matching kernel is using a generative model as a reference and obtain the similarity between the videos. A video is a sequence of frames which can be represented as a sequence of feature vectors and so hidden markov model is used as the generative model as it captures the stochastic information. The complete idea of this thesis can be described as building intermediate matching kernels using hidden markov model as generative model over which the SVM is used as a descriminative model for calssifying the actions based on the computed kernels. This idea is evaluated on the standard datasets like KTH, UCF50 and HMDB51

Contents

Chapter 1

Introduction

Human activity recognition has become an important area of research in computer vision. It has a wide range of applications in various domains like surveillance systems, sports video analysis, content based video retrieval, patient monitoring in medical domain, and in hybrid systems where a human needs to interact with electronic devices such as human-computer interfaces for gaming, tracking etc. In the above mentioned applications most of them require an automated system for recognizing human activities which might be either low level actions or the composition of several such low level actions. The class of action recognition problem can be arrnged in the increasing order of complexity as gesture, action, interaction and group activity. Gestures are simplest of movements with semantic importance. Actions are some what complex but limited to a single agent. Interactions are primitive activities comprising of multiple agents. From these categories simple action classification which consists of one instance of action is being chosen for this thesis

1.1 Issues In The Domain

The general approach in solving the problem of action classification is extracting desired features from the frames of the videos and use a classification algorithm based on the features used. Generally by using the training set of videos the system is learnt to classify an action into the corresponding class of action.In performing the above operation there might be the following issues which influence the choice of features to be extracted from the frames and the classification algorithm

1.1.1 Variations among and within the action classes

There are many actions with almost similar distribution of features but a human can only classify among these actions. For example, the movements between walking and jogging actions appear similar to a system and can easily differentiated by humans. Also there might be anthroprometric differences among the actors in different videos. Accounting to all these factors the system should be able to generalize over the actions belonging to a particular class and also differentiate among the other classes also. With the increasing number of actions to classify there is a high probability of misclassification among the classes. By using the domain knowledge related to the area of application one can train a system under constrained environment for achieving good accuracy of classification and there by the purpose of the application is fulfilled

1.1.2 Environment and Recording settings

Person localization can be very useful in recognizing the action performed by the actor. But the person localization depends a lot on the environment in which te action is being performed by the actor. Under certain constrained and static environment this problem seems to be easy but in some cluttered and dynamic environments the person localization is a difficult task and there by induces more complexity in recognizing the action performed by the actor. Illumination is also another factor which influences the action classification. Different view points produce different interpretations of the actions. So, care must be taken in accounting the camara positions of recording the actions. In addressing the action recognition problem these issues must be addressed so that a generic action classification system can be developed.

1.1.3 Obtaining and labelling of Training Data

On taking into account the above pointed issues it shows that training the system to classify actions is a much challenging step. Also, as the time progresses the video data keeps on piling and the diversity among the videos also increses. This further creates difficulty in carrying out the task. For this we require sufficient amount of diverse training data so that the system can learn from these training data effecively.

1.1.4 Varying pattern lengths

The human actions in videos last for different durations. The number of frames in which the actions are composed are different for different clips. This raises a challenge in comparing any two actions in videos. In this work this issue is being addressed and a HMM based Intermediate Matching Kernel is brought up for comparing two varying length patterns in particular classifying human actions in videos.

Chapter 2

Review of Previous work

2.1 Human Activity Recognition In Videos

Davis et al.[1] proposed two concepts namely MEI(Motion Energy Image) and MHI(Motion History Image) were used in recognizing the human activity. Binary MEI represents if there is an occurrence of an activity in an image sequence. Assumptions like the static background or the motion of the object can be separated from the camera. By using the MEI and the MHI the temporal templates are constructed for each of the considered actions. By using these temporal templates and some different matching approaches the unknown action is assigned a category.

Laptev et al.[2] used the application of the SVM over the set of local features was implemented for the action recognition. New dataset named as KTH was introduced in this work. Prior to this work several methods using global features extracted from video frames were used in recognizing the human activity. But these methods were dependent on recording conditions spatial resolution and the relative position with the camera. Hence the approach of using the local features was started with the intuition that the local measurements in terms of the spatiotemporal interest points. The local features were detected using the space scale representation using the Gaussian kernel.

Liu et.al[3] used the motion information from the video as the base for recognizing the human activity in the surveillance videos. For this the motion impression image is developed. The MII(motion Impression Image) is the combination of the two impression images namely Period Impression Image(PII) and Optical Flow Information Image(OFII) from different views. The PII is done by exploring the characteristics of the motion frequency and the OFII is obtained by the motion mode analysis. Next these constructed Impression Images are combined and quantized. These quantized impression images are used for the classification step. In the classification step we use the spatial pyramid matching based kernel(SPMK) based classifier for the recognition of the activity.

Ali et al.[4] Proposed the idea of using kinematic features obtained by the optical flow information from the videos for the human activity recognition in the videos. divergence, vorticity, symmetric and antisymmetric flow fields, second and third principal invariants of flow gradient and rate of strain tensor, and third principal invariant of rate of rotation tensor are the set of kinematic features under study. A spatiotemporal pattern can be obtained by computing each of the kinematic feature from the optical flow of a sequence of images. On these spatiotemporal patterns the Principal Component Analysis(PCA) is applied to get the kinematic modes. Next the procedure involves the classification step which is done by using the Multiple Instance Learning(MIL). Here each action video is represented as the bag of words of the kinematic modes. Each video is transformes into the kinematic mode space and the nearest neighbor algorithm is used for the classification.

Ahmed et al.[5] Combined Motion and the shape flow information for human activity recognition from multi view image sequences. Motion feature is obtained using the combined Local-Global optic flow. Global shape feature is extracted using the invariant moments with the flow deviations. So, from this the human action is represented as a set of multidimensional CLG optic flow and shape flow feature vectors in the spatialtemporal action boundary. The actions can be modeled as the multidiemensional HMMs for multiple views using the combined features.

Liu et al.[6] Proposed discriminative and action-based human action recognition model. In this the 3d interesting points cuboids are forms and these are used as the bag of spatiotemporal features to represent the videos. Theses cuboids are extracted by applying the linear filters in spatial and temporal directions. Now these cuboids are clustered into group of video words based on their appearance similarity. For this in the initial phase K-means algorithm is used and later the Maximization of Mutual Information is used so that it gives the optimal number of video word clusters automatically. Later SVM with histogram intersection kernel is used for classification.

Somayeh et al.[7] Proposed both unsupervised and supervised recognition of the human actions. In the unsupervised case, clustering the unlabeled video sequences into the same action based on the assumption that the number of clusters is priorly known. For this MMD(Maximum Mean Descripancy) along with the gaussian kernel. For the supervised case training an SVM model with a characteristic kernel is used. Generalized histogram intersection kernel and Histogram Characteristic kernels are used for the similarity measure. For the feature vector the harris descriptors on each frame are concatenated.

Yilmaz et al. [8] Mainly concentrated on the formation of the contour of the object by calculating the interested points in the 2D space. Next 3D spatiotemporal volume is obtained by taking all the 2d points along the time axis. Now this STV (Spatio Temporal Volume) can be considered as a 3D object in the (x,y,t) domain. Based on the differential geometric surface properties STVs are analyzed to identify action descriptors capturing spatial and temporal properties. Here each STV can be considered as a 3D rigid object. Now the matching of the actions becomes the object matching problem.

Liu et al.[9] explored the idea of "attributes" which are high-level semantic concepts. Each human action is defined as a composition of some basic semantic movements or attributes. These attributes, extracted from the KTH dataset were then applied as features to a latent SVM for classification

Yan Song et al.[10] proposed localized multiple kernel learning for realistic human action recognition. They have used static features from kef frames of shots and the dynamic spatiotemporal features. For localized multiple learning they have used the polynomial(homogeneous), polynomial(inhomogeneous) and Gaussian radial basis function kernels for each of the features.

Quoc V.Le et al.[11] came up with a robust feature for action recognition. Unlike SIFT or HOG they have extended the Independent Subspace Analysis algorithm to learn invariant spatio-temporal features from unlabeled video data. William Brendelet al. [12] modeled actions as time series of human postures. From the entire clip of video they have extracted multi scale regions from all the frames and constructed a sparse dictionary of most discriminative regions.

Sadanand et al. [13] introduced the Action Bank which comprises of a bank of template videos which are used to compute the feature descriptors for the input video. The implementation of Action Bank used comprised of 205 manually selected template videos consisting of 56 different actions. For each template in the bank, we determine the correlations of various volumes in the video with the template and this comprises of the the feature descriptor for the input video.

If there are N_a detectors in the action bank and each action detector is run at N_a scales we have N_aXN_a correlation volumes. On this volumetric max pooling is applied and three levels of octree are taken.Each action-scale pair gives 1^3+2^3+4^3 i.e 73 diemensional vector.The total length of the action bank feature vector is N_aX N_sX73. They have used 205 templates at one scale and so, 205x1x73 = 14965 length feature vector for each clip of video is generated

Wang et al. [14] used motion boundary descriptors and dense trajectories are used to represent a video. By this they have captured the local motion and the dense representation of the foreground motion and the surroundings also. The dense trajectories are extracted using the traditional optical flow algorithm. Along with the usage of HOG(Histograms of Oriented Gradients) HOF(Histograms Of Optical flow) they have introduced a descriptor MBH(Motion Boundary Histogram) which rely on differential optical flow.

2.2 Kernel Methods For Varying Length Pattern Analysis

A varying length pattern can be represented either as a set of feature vectors or as a sequence of feature vectors. For example in scene image classificaiton problem while representing an image by segmenting it, the allignment of the feature vectors in the representation doesnot matter. It is required if an object is present in an image or not rather than its position. This representation can be considered as a set of feature vectors. Coming to videos, a video is represented as a sequence of frames. In this case when a feature vector is used to represent a frame we have a sequence of feature vectors representing a video.

As in [15] Classification of varying length patterns can be done using two approaches.

Approach 1:

- Varying length pattern is mapped onto a fixed length pattern
- A kernel for fixed length patterns such as Gaussian kernel is used to find similarity between any two patterns

Approach 2:

- A suitable kernel for varying length patterns is designed
- Kernel functions designed for varying length patterns are known as dynamic kernels

In this thesis a study of constructing the dynamic kernels for both set and sequence of local feature vectors has been made.From [15] There are different ways for building up the dynamic kernels.They are as follows

- Explicit mapping based approaches:Explicit mapping based approaches map a set of local feature vectors on to a fixed diemensional representation.In this fixed dimension a kernel is defined for finding out similarity beween any two set of feature vectors.
- Probabilistic distance metric based approaches:Probabilistic distance metric based technique uses probabilistic distributions for variable length patterns and find kernels to compute distance between the probability distributions. Example GMM super vector kernel,Probability product kernel.
- Dynamic time alignment based approaches:Dynamic time alignment approaches use algorithms like dynamic time warping for calculating distance between two sequence of feature vectors. This approach is used in tha applications where the sequence of the feature vectors matters most in finding the similarity.
- Matching based approaches: Matching based kernels first matches the set of local feature vectors pairwise and then the kernel function is computed as either taking the sum of similarities between all the pairs or taking the sum of similarities between the pairs that are matched with high similarity. The summation kernel and mathcing kerenl are examples for this approach

2.2.1 Fisher Kernel

Fisher Kernel[16] is used for finding out the similarity between patterns represented as set of feature vectors. Fisher Kernel exploits the generative model GMM and maps the sets of local feature vectors into a fixed dimensional Fisher Score Vector. This fixed dimensional space is defined as the GMM based likelihood score space. The likelihood score space can be formed by using the derivative of the log-likelihood with respect to the parameters of the GMM. For a set of feature vectors, each of the feature vectors is of dimension d, the vector of gradient of the log-likelihood with respect to mean of each component of the GMM is computed. In the Fisher score space a set of local feature vectors can be represented in a fixed dimension as a super vector of all the Fisher Score Vectors of each of the components.For a set of local feature vectors $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ...\mathbf{x_T}]$ the log-likelihood gradient vector with respect to the mean $\mu_\mathbf{q}$ of a q^{th} component of a GMM is given by

$$\mathbf{\Psi}_\mathbf{q}^{(\mu)} = \sum_{\mathbf{t=1}}^{\mathbf{T}} \gamma_\mathbf{q}(\mathbf{x_t})\mathbf{z_{tq}} \tag{2.1}$$

where $\mathbf{z_{tq}} = \mathbf{\Sigma_q^{-1}}(\mathbf{x_t} - \mu_\mathbf{q})$

From the gradient vectors of the set of local vectors with respect to the $\mu_\mathbf{q}, \sigma_\mathbf{q}, \mathbf{w_q}$ of a particular component q, of a GMM the Fisher vector is written as the super vector as follows

$$\mathbf{\Phi_q}(\mathbf{X}) = [\mathbf{\Psi}_\mathbf{q}^{(\mu)}, \mathbf{\Psi}_\mathbf{q}^{(\mathbf{\Sigma})}, \mathbf{\Psi}_\mathbf{q}^{(\mathbf{w})}]^\mathbf{T} \tag{2.2}$$

By using all these Fisher score vectors of all the Q components of a GMM the set of local vectors can be represented in the fixed dimensional supervector Φ_{FK} as follows

$$\mathbf{\Phi_{FK}(X) = [\Phi_1(X)^T, \Phi_2(X)^T, \Phi_3(X)^T, ...\Phi_Q(X)^T]} \tag{2.3}$$

The Fisher score vector has a dimension of $D = Q(1 + d + d^2)$. The Fisher kernel between two sets of local feature vectors $\mathbf{X_m} and \mathbf{X_n}$ can be written as

$$\mathbf{K_{FK}(X_m, X_n) = \Phi_{FK}(X_m)^T F^{-1} \Phi_{FK}(X_n)} \tag{2.4}$$

Here F is the Fisher information matrix written as Fisher Information Matrix

$$\mathbf{F = \frac{1}{L} \sum_{l=1}^{L} \Phi_{FK}(x_l) \Phi_{FK}(x_l)^T} \tag{2.5}$$

2.2.2 Probabilistic Sequence Kernel

In Probabilistic Sequence Kernel[17], the local feature vectors are mapped onto a fixed dimensional probabilistic feature vector using GMM and Universal Background Model(UBM). Each of the GMM and the UBM are modeled with Q components. In this higher dimension a local feature vector is defined as its responsibility towards each of the Q components in both the models. The dimension in this higher space is $2Q$ for each of the local feature vector. This vector is called as the probabilistic alignment vector. $\mathbf{\Psi}(X) = [\gamma_1(x), \gamma_2(x), ..., \gamma_{2Q}(x)]^T$ represents a probabilistic alignment vector in the fixed diemension $2Q$ for a local feature vector x.A fixed diemensional vector $\mathbf{\Phi_{PSK}}(X)$ for a set of local feaure vectors $X = [x_1, x_2, x_3, ...x_t]$ can be written as

$$\mathbf{\Phi_{PSK}}(X) = \frac{1}{T} \sum_{t=1}^{T} \mathbf{\Psi}(\mathbf{X_t}) \tag{2.6}$$

For any two set of feature vectors $X_m = [x_{m1}, x_{m2}, x_{m3}, ...x_{mT_m}]$,$X_n = [x_{n1}, x_{n2}, x_{n3}, ...x_{nT_n}]$ the pyramid sequence kernel can be computed as

$$\mathbf{K_{PSK}(X_m, X_n) = \Phi_{PSK}(X_m)^T} S^{-1} \mathbf{\Phi_{PSK}(X_n)^T} \tag{2.7}$$

Here, S is the correlation matrix taken as

$$\mathbf{S = \frac{1}{M} R^T R} \tag{2.8}$$

where the matrix R is formed by taking probabilistic alignment vectors of the set of local feature vectors as the rows and M is the total number of local feature vectors in the training data.

2.2.3 GMM Supervector Kernel

In GMM Supervector Kernel[18], the mapping is done from the set of local feature vectors to the corresponding GMM supervectors in the higher fixed dimensional space. An example-specific adapted GMM is built for each example by adapting the means of the UBM using the data of that example.For an example $\mathbf{X} = [x_1, x_2, x_3, ...x_T]$, $\mu_{\mathbf{q}}^{(\mathbf{x})}$ represent the mean vector of q^{th} component in example-specific adapted GMM, its GMM vector $\mathbf{\Psi_q(X)}$ can be obtained as

$$\mathbf{\Psi_q(X)} = [\sqrt{\mathbf{w_q}}\mathbf{\Sigma_q^{\frac{-1}{2}}}\mu_{\mathbf{q}}^{(\mathbf{X})}]^{\mathbf{T}} \tag{2.9}$$

The GMM supervector $\mathbf{\Psi_{GMMSV}(X)}$ is given by

$$\mathbf{\Psi_{GMMSV}(X)} = [\mathbf{\Psi_1(X)^T}, \mathbf{\Psi_2(X)^T}, \mathbf{\Psi_3(X)^T}...\mathbf{\Psi_Q(X)^T}]^{\mathbf{T}} \tag{2.10}$$

This GMM supervector has a dimension of $D = Qd$. The GMM supervector kernel for a pair of example X_m and X_n is given as

$$\mathbf{K_{GMMSV}(X_m, X_n)} = \mathbf{\Phi_{GMMSV}(X_m)^T\Phi_{GMMSV}(X_m)} \tag{2.11}$$

2.2.4 Summation kernel

Summation kernel[19] belongs to the class of matching based approach kernels.For any two sets of local feature vectors, the summation kernel first computes the similarity between all the pairs local feature vectors from both the sets by using any base kernel function. Next, all these similaritise are summed up to give the similarity between two sets of local feature vectors.The summation kernel between two sets of local feature vectors $\mathbf{X_m} = [\mathbf{X_{m1}}, \mathbf{X_{m2}}, ...\mathbf{X_{mT_m}}]$ and $\mathbf{X_m} = [\mathbf{X_{n1}}, \mathbf{X_{n2}}, ...\mathbf{X_{nT_n}}]$ is given as

$$\mathbf{K_{SK}(X_m, X_n)} = \sum_{\mathbf{t=1}}^{\mathbf{T_m}}\sum_{\mathbf{t'=1}}^{\mathbf{T_n}} \mathbf{K(x_{mt}, x_{nt'})} \tag{2.12}$$

The summation kernel is a positive semi difinite kernel as it is a sum of positive semi definite kernels. The total number of comparisions made in the summation kernel is equal to the product of the cardinalities of the two sets of local feature vectors.For two sets of local feature vectors $\mathbf{X_m} = [\mathbf{X_{m1}}, \mathbf{X_{m2}}, ...\mathbf{X_{mT_m}}]$ and $\mathbf{X_m} = [\mathbf{X_{n1}}, \mathbf{X_{n2}}, ...\mathbf{X_{nT_n}}]$ the number of comparisions made is $T_m * T_n$. This is computionally very high for sets with large number of feature vectors.

2.2.5 Matching Kernel

Matching kernel[20] also belongs to the matching based approach famiy of kernels. Unlike summing up the similarities among all the pairs of local feature vectors in summaton kernel, each local feature vector in a set of local feature vector is matched with the most similar local feature vector from the other set of local feature vectors. The sum of the similarites of all the pairs of matched local feature vectors from both the sets gives the matching kernel. The summation kernel between two sets of local feature vectors $\mathbf{X_m} = [\mathbf{X_{m1}}, \mathbf{X_{m2}}, ...\mathbf{X_{mT_m}}]$ and $\mathbf{X_m} = [\mathbf{X_{n1}}, \mathbf{X_{n2}}, ...\mathbf{X_{nT_n}}]$ is given as

$$\mathbf{K_{MK}(X_m, X_n)} = \sum_{\mathbf{t=1}}^{\mathbf{T_m}} \max_{\mathbf{t'}} \mathbf{k(x_{mt}, x_{nt'})} + \sum_{\mathbf{t'=1}}^{\mathbf{T_n}} \max_{\mathbf{t}} \mathbf{k(x_{mt}, x_{nt'})} \tag{2.13}$$

For two sets of local feature vectors $\mathbf{X_m} = [\mathbf{X_{m1}}, \mathbf{X_{m2}}, ...\mathbf{X_{mT_m}}]$ and $\mathbf{X_m} = [\mathbf{X_{n1}}, \mathbf{X_{n2}}, ...\mathbf{X_{nT_n}}]$ the number of comparisions made for matching kernel is $2 * T_m * T_n$.

As both matching and summation kernels are of order $\mathcal{O}(T^2)$ in complexity, where T is the maximum cardinality between the two sets of local feature vectors, these kernels are not computationally feasible.

2.2.6 Intermediate Matching Kernel

As the computation of summation and matching kernels is highly intensive boughorbel et.al proposed the use of Intermediate Matching Kernel(IMK)[21]. The idea used was first select a Q number of virtual feature vectors. Next for each of the Q virtual feature vectors find out a closest local feature vector from each of the sets of local feature vectors.Now the base kernel between local feature vectors from two sets of local features which are closest to a particular virtual feature vector is computed. Now there are Q such base kernels. The IMK between the two sets of local feature vectors is the sum of Q base kernels. Let $\mathbf{X_m} = [\mathbf{X_{m1}}, \mathbf{X_{m2}}, ...\mathbf{X_{mT_m}}]$ and $\mathbf{X_m} = [\mathbf{X_{n1}}, \mathbf{X_{n2}}, ...\mathbf{X_{nT_n}}]$ be two sets of local feature vectors and $\mathbf{V} = [\mathbf{V_1}, \mathbf{V_2}, ...\mathbf{V_Q}]$ be the set of virtual feature vectors. For a q^{th} virtual feature vector V_q two local feature vectors ${x_m}_q^*$,${x_n}_q^*$ closest to V_q are chosen from X_m and X_n respectively.

$$\mathbf{X_m}_\mathbf{q}^* = \mathbf{arg}\min_{\mathbf{x \epsilon X_m}} D(x, v_q) and \mathbf{X_n}_\mathbf{q}^* = \mathbf{arg}\min_{\mathbf{x \epsilon X_n}} D(x, v_q) \tag{2.14}$$

where D is a distance metric for computing distance between local feature vectors of X_m or X_n to a virtual feature vector from V. Now, the IMK between X_m and X_n can be written as

$$\mathbf{K_{IMK}}(\mathbf{X_m}, \mathbf{X_n}) = \sum_{\mathbf{q=1}}^{\mathbf{Q}} \mathbf{k}(\mathbf{x_m}_\mathbf{q}^*, \mathbf{x_n}_\mathbf{q}^*) \tag{2.15}$$

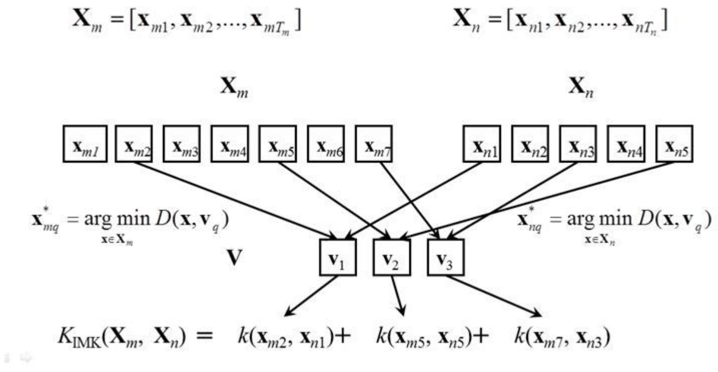
$\mathbf{X}_m = [\mathbf{x}_{m1}, \mathbf{x}_{m2}, ..., \mathbf{x}_{mT_m}]$
$\mathbf{X}_n = [\mathbf{x}_{n1}, \mathbf{x}_{n2}, ..., \mathbf{x}_{nT_n}]$
$\mathbf{X}_m$
$\mathbf{X}_n$
$\mathbf{x}_{m1}$ $\mathbf{x}_{m2}$ $\mathbf{x}_{m3}$ $\mathbf{x}_{m4}$ $\mathbf{x}_{m5}$ $\mathbf{x}_{m6}$ $\mathbf{x}_{m7}$
$\mathbf{x}_{n1}$ $\mathbf{x}_{n2}$ $\mathbf{x}_{n3}$ $\mathbf{x}_{n4}$ $\mathbf{x}_{n5}$
$\mathbf{x}_{mq}^* = \arg\min_{\mathbf{x}\in\mathbf{X}_m} D(\mathbf{x}, \mathbf{v}_q)$
$\mathbf{x}_{nq}^* = \arg\min_{\mathbf{x}\in\mathbf{X}_n} D(\mathbf{x}, \mathbf{v}_q)$
$\mathbf{V}$
$\mathbf{v}_1$ $\mathbf{v}_2$ $\mathbf{v}_3$
$K_{\mathrm{IMK}}(\mathbf{X}_m, \mathbf{X}_n) = k(\mathbf{x}_{m2}, \mathbf{x}_{n1}) + k(\mathbf{x}_{m5}, \mathbf{x}_{n5}) + k(\mathbf{x}_{m7}, \mathbf{x}_{n3})$

Figure 2.1: Intermediate Matching Kernel

The number of calculations of D, the distance between local feature vector and virtual feature vector for finding out one closest pair of local feature vectors from two sets is $(T_m + T_n)$. For finding out Q such pairs it takes $Q(T_m + T_n)$ calculations. The total complexity ca be written as $O(T)$, where T is the cardinality of the largest set of local feature vectors. This cost is less when compared to the computaional cost of matching and summation kernels.If Q is much smaller compares to T_m or T_n, the computaional cost reduces by a huge margin and this makes construction of IMK more feasible than the matching and summation kernels. For constructing the IMKs the next question that rises is the selsction of set of virtual feature vectors V. The following figure represents the idea of IMK.

In [21] the the training data is clustered into Q number of clusters and the centers obtained for each of the cluster are taken as the virtual feature vectors. As the base kernel, gaussian kernel was used to obtain the closest pair of local feature vectors to the virtual feature vectors from the set of local feature vectors. This approach is called as Codebook Based IMK(CBIMK) as the centers of the clusters are used to build the codebook.

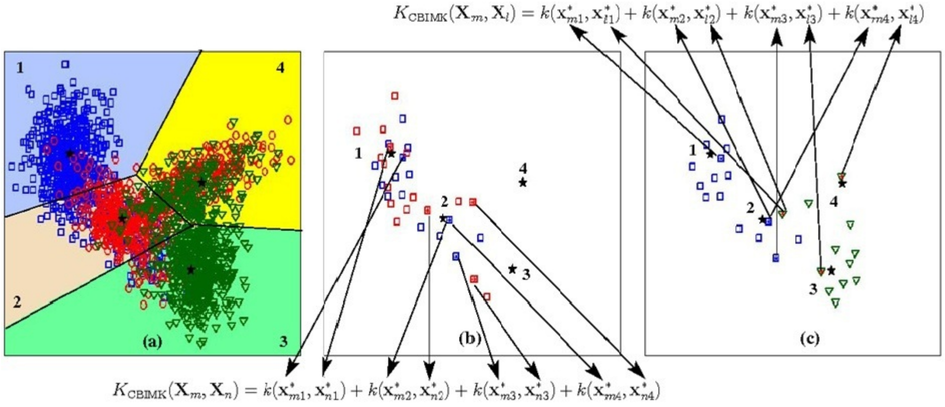
$K_{\text{CBIMK}}(\mathbf{X}_m, \mathbf{X}_l) = k(\mathbf{x}^*_{m1}, \mathbf{x}^*_{l1}) + k(\mathbf{x}^*_{m2}, \mathbf{x}^*_{l2}) + k(\mathbf{x}^*_{m3}, \mathbf{x}^*_{l3}) + k(\mathbf{x}^*_{m4}, \mathbf{x}^*_{l4})$
1
4
2
3
(a)
1
4
2
3
(b)
1
2
4
3
(c)
$K_{\text{CBIMK}}(\mathbf{X}_m, \mathbf{X}_n) = k(\mathbf{x}^*_{m1}, \mathbf{x}^*_{n1}) + k(\mathbf{x}^*_{m2}, \mathbf{x}^*_{n2}) + k(\mathbf{x}^*_{m3}, \mathbf{x}^*_{n3}) + k(\mathbf{x}^*_{m4}, \mathbf{x}^*_{n4})$

Figure 2.2: Intermediate Matching Kernel

2.2.7 GMM based IMK

CIGMMIMK[15]: From the CBIMK it is observed that the kernel computation is based on the centers of the clusters rather than the entire distribution of the data available. To exploit the distribution of the data [15] proposed the use of GMM instead of clustering the training data as om CBIMK. For that a class independant GMM with Q components is constructed over all the available data. This brings into picture the use of mean vectors,co-variance matrices and mixture coefficients of all the components in the GMM for representing the virtual feature vectors. As the distance metric used in CBIMK for finding the closest local feature vector from a virtual feature vector, in the GMM based IMK the responsibility term of the q^{th} component for a local feature vector is used to find the closest local feature vector for a component. If the GMM is modelled with Q components we have total of 2Q local feature vectors closest to the Q components from both set of local feature vectors. Q pairs of base kernels are computed with respect to Q virtual feature vectors. The local feature vectors from both the sets are seleced as follows

$$\mathbf{X_m}^*_{\mathbf{q}} = \mathbf{arg}\max_{\mathbf{x}\epsilon\mathbf{X_m}} \gamma_{\mathbf{q}}(\mathbf{x}) and \mathbf{X_n}^*_{\mathbf{q}} = \mathbf{arg}\max_{\mathbf{x}\epsilon\mathbf{X_n}} \gamma_{\mathbf{q}}(\mathbf{x}) \tag{2.16}$$

where $\gamma_q(x)$ is the responsibility term for the q^{th} component. This is called as Class Independant GMM based Intermediate Matching Kernel as the GMM is constructed from the data of all calsses.

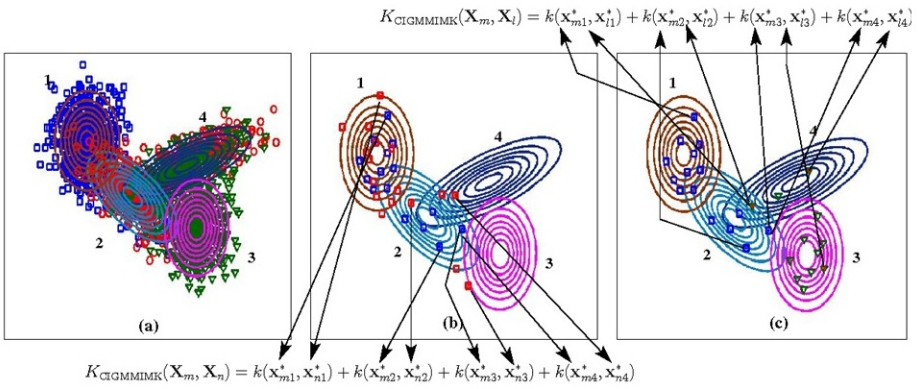
$K_{\mathrm{CIGMMIMK}}(\mathbf{X}_m, \mathbf{X}_l) = k(\mathbf{x}^*_{m1}, \mathbf{x}^*_{l1}) + k(\mathbf{x}^*_{m2}, \mathbf{x}^*_{l2}) + k(\mathbf{x}^*_{m3}, \mathbf{x}^*_{l3}) + k(\mathbf{x}^*_{m4}, \mathbf{x}^*_{l4})$
1
4
2
3
(a)
1
4
2
3
(b)
1
4
2
3
(c)
$K_{\mathrm{CIGMMIMK}}(\mathbf{X}_m, \mathbf{X}_n) = k(\mathbf{x}^*_{m1}, \mathbf{x}^*_{n1}) + k(\mathbf{x}^*_{m2}, \mathbf{x}^*_{n2}) + k(\mathbf{x}^*_{m3}, \mathbf{x}^*_{n3}) + k(\mathbf{x}^*_{m4}, \mathbf{x}^*_{n4})$

Figure 2.3: Intermediate Matching Kernel

2.2.8 Kernels for Sequence Of feature vectors

As seen earlier the kernels for sets of local feature vectors exploit the genertive model GMM for mapping local feature vectors into some other fixed dimensional space.For the sequence of feature vectors the Hidden Markov Model(HMM)[22] is used as the generative model so that the sequential informatiom among the sequence of vectors is preserved. Under this mapping approach [23] proposed fisher score space as the kernel feature space for sequence of feature vectors. Based on the likelihood from the HMM the sequence of vectors is represented as the Fisher Score Vector. The derivative of the log-likelihood of the sequence of feature vectors with respect to the HMM parameters and the state specific GMM parameters is considered as the score space. The Fisher Score Vector can be defined as the super vector of Fisher Kernel based on GMM for each of the state. The Fisher Kernel is computed by taking a single calss HMM into consideration. So the classification using Fisher Kernel with SVM is a oneVSrest approach. [24] proposed another score space namely, likelihood ratio kernel as the Kernel Space.Here, the kernel space is defined as the derivative of the ratio of log-likelihood of sequence of feature vectors with respect to HMM models of two classes. The rest of the kernel computaion is same as the Fisher Kernel. The only difference between the Fisher Kernel and the likelihood ratio kernel is the kernel feature space. In Fisher kernel score space is computed for every class, where as in likelihood ratio kernel score space is compared for every pair of classes. [25] proposed probability product kernel where each sequence is represented as a HMM and Kullback-Leibler(KL) divergence is used as the dissimilarity measure between the two HMMs. By kernelizing this dissimilarity values the probability product kernel is obtained.

Chapter 3

Action Recognition Using HMM Based IMK

Until now many action recognition ideas have been proposed and most have them have succeeded in delivering good performance on the controlled datasets. But, most of them lag in classification human actions in videos which are shot wild without any restrictions on recording environment and actions. The issue of the actions being varied in length have not been addressed effectively in the previous work. Features like action bank have succeded in eliminating the impact of variable length pattern in representation but have not produced effective results on the complicated datasets. For addressing the issue of varibale length pattern classification in videos the technique of building up the dynamic kernels was chosen in this thesis. In this thesis the idea of using Complete Sequence HMM based IMK(CSHIMK)[26] with Support Vector Machine(SVM) is proposed for Human Action Classification In Vdeos.

3.1 HMM Based IMK

A.D Dileep.et al[26] proposed Complete Sequence HMM Intermediate Matching Kernel(CSHIMK) for computing similarity between sequences of local feature vectors. In this approach intermediate matching kernel is computed by taking HMM as the generative model. First for each of the available classes the continuous density HMM is built. In this each of the states is modeled as a guassian mixture model with Q number of mixtures. From each of the sequences of the local feature vectors, a local feature vector with highest responsibility term for a particular component in a particular state is selected.Suppose we have a sequence of local feature vectors $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}, ...\mathbf{x_T}]$, a continuous density HMM with λ as the set of parameters, we have at a particular time t, the probability of being in a state i as follows.

$$\mathbf{V_{it}} = \mathbf{P}[\mathbf{s_t} = \mathbf{i}|\mathbf{X}, \lambda] \tag{3.1}$$

Let $\gamma_q(x_t)$ represent the responsibility term showing how much the component q is contributing for the local feature vector x_t. For selecting the local feature vector in a particular state corresponding to a particular component in a GMM th responsibility term is calculated as follows

$$\mathbf{R_{iq}(x_t, X|\lambda) = v_{it}\gamma_{iq}(x_t)} \tag{3.2}$$

where v_{it} represents the probability of a local feature vector x_t being in a state i and $\gamma_{iq}(x_t)$ represents the contribution pf q^{th} component of state i for the local feature vector x_t From each of the sequence of local feature vectors,the local feature vector with highest responsibility term are selected for computing the base kernels with respect to each component of a GMM in a particular state. Let X_m and X_n represent two sequences of local feature vetors, λ be the set of parameters of a HMM, $x^*{}_{miq}$,$x^*{}_{niq}$ represent the local feature vectors selected from X_m,X_n respectively from state i and q^{th} component of the GMM in state i

$$\mathbf{x^*{}_{miq} = arg \max_{x_t \epsilon X_m} R_{it}(x_t|X_m, \lambda)} and \mathbf{x^*{}_{niq} = arg \max_{x_t \epsilon X_n} R_{it}(x_t|X_n, \lambda)} \tag{3.3}$$

In th CSHIMK within the state a state specific GMM based IMK is computed among the sets of the local feature vectors selected fron the sequences of feature vectors in that particular state. The sum of the state specific GMM based IMKs of all the states gives the CSHIMK between two sequences of local feature vectors. This can be written as

$$\mathbf{K_{CSHIIMK} = \sum_{i=1}^{N} \sum_{q=1}^{Q_i} k(x^*{}_{miq}, x^*{}_{niq})} \tag{3.4}$$

The CSHIMK is a valid positive semi-definite kernel if the base kernel is taken as the positive semi-definite kernel.

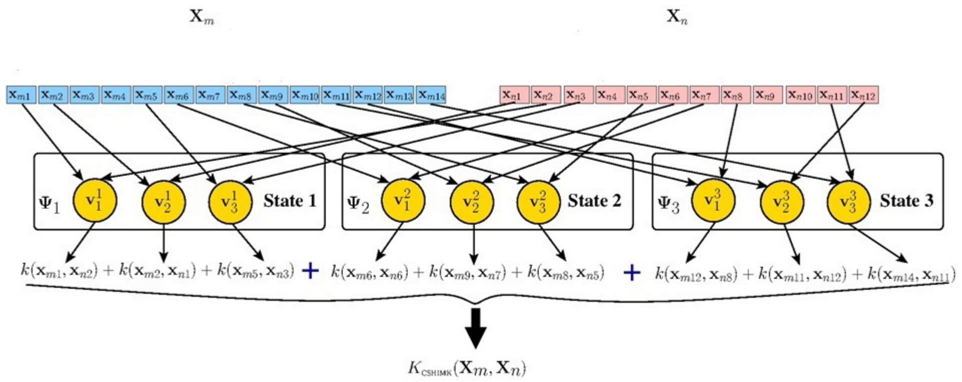
$\mathbf{x}_m$
$\mathbf{x}_n$
$\mathbf{x}_{m1}$ $\mathbf{x}_{m2}$ $\mathbf{x}_{m3}$ $\mathbf{x}_{m4}$ $\mathbf{x}_{m5}$ $\mathbf{x}_{m6}$ $\mathbf{x}_{m7}$ $\mathbf{x}_{m8}$ $\mathbf{x}_{m9}$ $\mathbf{x}_{m10}$ $\mathbf{x}_{m11}$ $\mathbf{x}_{m12}$ $\mathbf{x}_{m13}$ $\mathbf{x}_{m14}$
$\mathbf{x}_{n1}$ $\mathbf{x}_{n2}$ $\mathbf{x}_{n3}$ $\mathbf{x}_{n4}$ $\mathbf{x}_{n5}$ $\mathbf{x}_{n6}$ $\mathbf{x}_{n7}$ $\mathbf{x}_{n8}$ $\mathbf{x}_{n9}$ $\mathbf{x}_{n10}$ $\mathbf{x}_{n11}$ $\mathbf{x}_{n12}$
Ψ_1 $\mathbf{v}_1^1$ $\mathbf{v}_2^1$ $\mathbf{v}_3^1$ State 1
Ψ_2 $\mathbf{v}_1^2$ $\mathbf{v}_2^2$ $\mathbf{v}_3^2$ State 2
Ψ_3 $\mathbf{v}_1^3$ $\mathbf{v}_2^3$ $\mathbf{v}_3^3$ State 3
$k(\mathbf{x}_{m1}, \mathbf{x}_{n2}) + k(\mathbf{x}_{m2}, \mathbf{x}_{n1}) + k(\mathbf{x}_{m5}, \mathbf{x}_{n3})$ + $k(\mathbf{x}_{m6}, \mathbf{x}_{n6}) + k(\mathbf{x}_{m9}, \mathbf{x}_{n7}) + k(\mathbf{x}_{m8}, \mathbf{x}_{n5})$ + $k(\mathbf{x}_{m12}, \mathbf{x}_{n8}) + k(\mathbf{x}_{m11}, \mathbf{x}_{n12}) + k(\mathbf{x}_{m14}, \mathbf{x}_{n11})$
$K_{\text{CSHIMK}}(\mathbf{X}m, \mathbf{X}n)$

Figure 3.1: Complete Sequence HMM based Intermediate Matching Kernel

The above figure explains the computaion of CSHIMK between two sequences of local feature vectors X_m and X_n, where the HMM is a three state model and the state specific GMMs are of three mixtures.In each of the states for calculating the posterioir probability of each of the components for the local feature vectors the number of computations required is $Q * (T_M + T_n)$. This can be written as $Q * T$ where T is the maximum cardinality between the two sequences. So, the total computaions required for selecting the local feature vectors across all the states of a HMM can be written as $\mathbf{\sum_{i=1}^{N} Q * (T_M + T_n)}$. So, the computaional complexity for the CSHIMK is in the order of $O(NQT)$, where T is the maximum length of the sequences T_m and T_n.

3.2 Features And Approach

In this thesis Histogram Of Oriented Gradients(HOG)[27] descriptor is proposed to be used as the feature. In the proposed approach the hog desciptors are extracted per frame from all the videos of all classes and these features when combined form a sequence of local feature vectors, each sequence representing a video. HMMs are constructed for each of the class of action and each state is modeled as a GMM. Like Fisher Kernel in CSHIMK also we use HMM of a particular class for computing the kernels between the sequences of local feature vectors of same class and sequences of other classes also. So, the SVM that is developed for classifying the sequences uses the oneVSrest approach.

By using CSHIMK the kernels are computed for each training sequence of local feature vectors of a class with all the sequences in training data of the same class and also the rest of the classes and written as a super vector of these similarities in the kernel space. Similarly the kernels are computed for the test sequences also with all the training sequences from all other classes. The SVM is trained with the train kernels and the test sequences are classified in the oneVSrest manner.

Chapter 4

Experiments And Results

For the evaluation of the proposed approach experiments were conducted on standard datasets like KTH[2],UCF11[28],UCF101[29] and hmdb51[30].

4.1 KTH dataset

[2] proposed the KTH datset. It consists of six actions namely walking,runnning.jogging,hand-waving,hand-clapping and boxing. Each action is performed by 25 different subjects(actors) under four different environment settings. These settings include outdoor actions,outdoor actions varying in scale, indoor actions, actions outdoor with different clothes and view points. On a total each class has a total of 100 clips of videos. The background is static and only single actor is present in the video. The following figure shows the examples of actions in KTH dataset. Among the four different scenarios the clips of one scenario are used for testing and the rest of the three for training. HOG features were extracted from all the videos and each video is represented as the sequence of local feature vectors. The HMMs are constructed using the training samples and the training and testing kernels were built in the oneVsRest approach. Experiments were conducted by varying th number of states of HMMS in the window 2,3,4,5,6,7 with two mixtures per state. From these experiments it was observed that the seven state model with two mixtures per state gave the best performance.

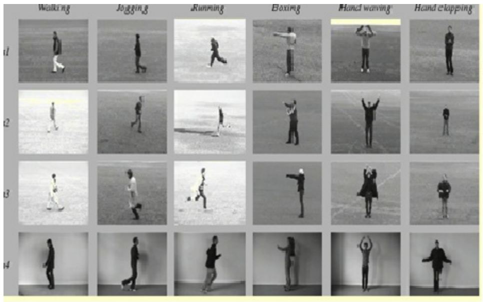

Figure 4.1: KTH Datset

Action	Walking	Jogging	Running	Boxing	Waving	Clapping
Walking	25	0	0	0	0	0
Jogging	0	25	0	0	0	0
Running	0	0	25	0	0	0
Boxing	0	0	0	25	0	0
Waving	0	0	0	0	25	0
Clapping	0	0	0	0	2	23

4.2 UCF11

Jingen et.al[28] introduced the UCF11 dataset also called as UCF Youtube action dataset. It contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horse back riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. Each class contains 25 groups of videos and in each group a minimum of four clips are present. In a group almost all the clips have common features like same actor,same environment, but differ in view points. This dataset is the first challenging dataset as the vidoes are taken from the youtube and are shot in wild. The actions included in the dataset are shown below.

Figure 4.2: UCF11 Datset

4.2.1 Results

The overall accuracy on the UCF-11 dataset obtained is 84.98%

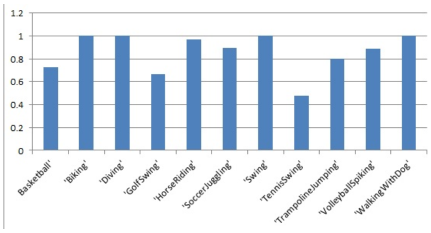

Figure 4.3: UCF11 Datset

4.3 UCF101

Soomro et.al[29] introduced the UCF101 dataset. This dataset is the most diversified dataset with a set of 101 action classes and contains most of the normal human actions that we witness in the daily life. The videos are also have large differences in object view, backgrounds, clutterdness, objectscale. This dataset is an advancement of the UCF11 dataset. As in UCF11 dataset here akso we have each class of action having 25 groups of videos with 4-7 clips in each group. In a group almost all the clips have common features like same actor,same environment, but differ in view points.The categories of actions that this dataset has addressed can be placed among the five groups like

- Human-Object Interaction
- Body-Motion Only
- Human-Human Interaction
- Playing Musical Instruments Sports

The actions invoved in the UCF101 dataset are shown in the following figure.

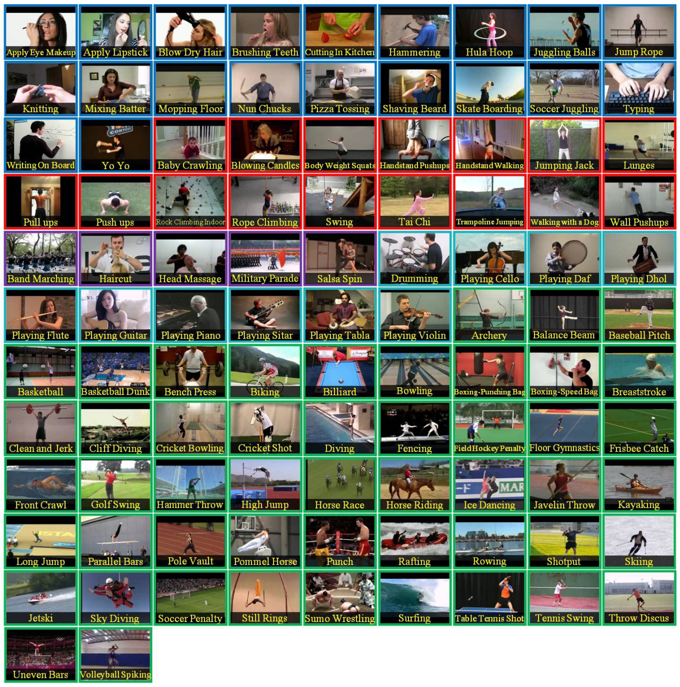

Figure 4.4: UCF101 Datset

4.3.1 Results

The overall accuracy obtained of the UCF-101 dataset is 63.1%

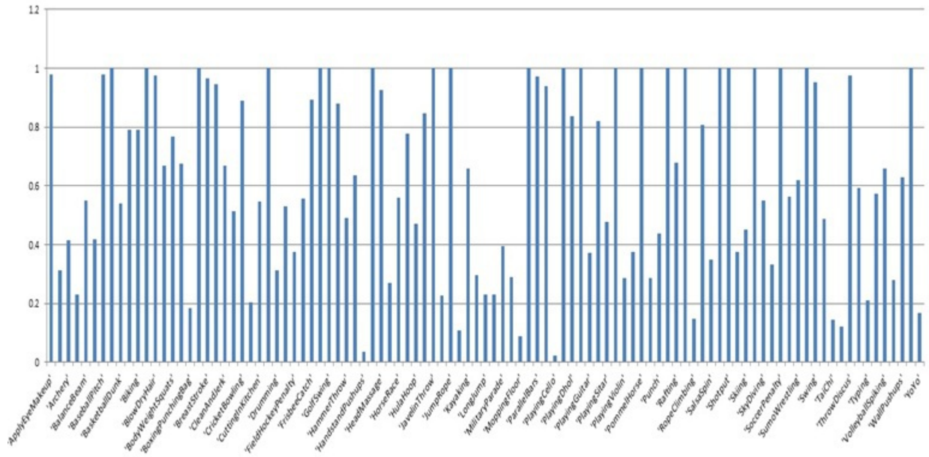

Figure 4.5: UCF101 Results

4.4 HMDB51

The HMDB51[30] dataset consists of 51 classes of action. On a total there are 6849 clips with each class having a minimum of 101 clips. These actions include the categories of general facial expressions like smile,chuckle,laugh, face actions with objects,general body movements, body movements with object interactions like brushing hair, pushing something,riding etc and the human interactions like fencing,hug etc collected from youtube, google videos. This dataset is most complicated than the other available datasets as the clips are shot without any restrictions and doesn't have much variations among some classes.

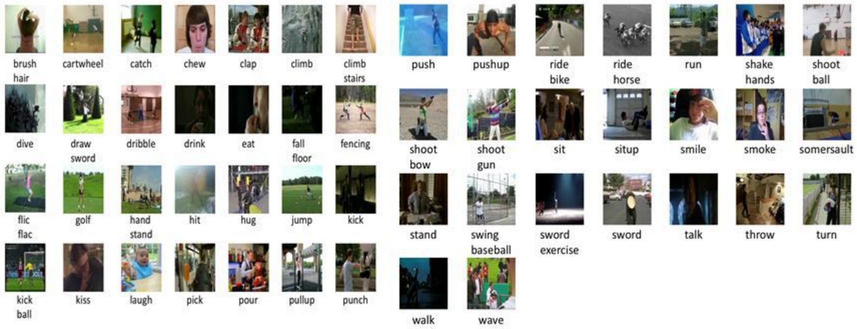

Figure 4.6: HMDB51 Datset

4.4.1 Results

The dataset contains of three splits of training and testing samples. Experiments were conducted on all the splits and the best are presented in the figure 4.7. Overall accuracy:55.1%

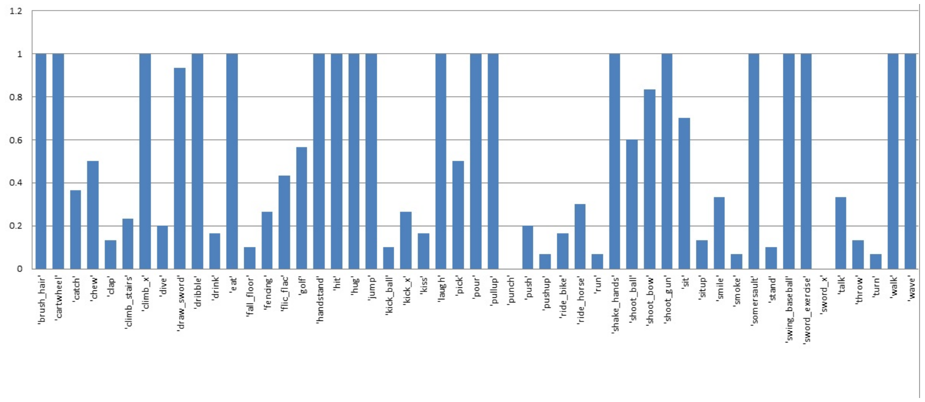

Figure 4.7: HMDB51 result

Chapter 5

Conclusion And Future Work

In this thesis a brief study about dynamic kernels for varying length pattern matching has been done and the HMM based intermediate matching kernel was used for classifying human actions in videos. The idea of using svm based classifier with HMM based IMK was evaluated using several standard datasets like KTH,UCF11,UCF101,hmdb51 and results have been proved to be promising and nearly surpassed the state of art results. It can be also noticed that using the posterior probability weighted dynamic kernels can be constructed which might be more accurate in the classification. It is also observed that the computation time for the construction of the HMM based IMK is very less when compared to other dynamic kernels like Fisher kernel,Summation kernel etc. Experiments have been done considering only HOG as the feature vector, furthur run of experiments using the high level feature like Motion Boundary Histogram(MBH) which in recent times proved to be most useful might increase the classification accuracy.

References

[1] J. Davis and A. Bobick. The representation and recognition of human movement using temporal templates. In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on. 1997 928–934.

[2] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, volume 3. 2004 32–36 Vol.3.

[3] J. Liu, T. Zhang, H. Lu et al. Human action recognition in videos using motion impression image. In Proceedings of the First International Conference on Internet Multimedia Computing and Service. ACM, 2009 174–178.

[4] S. Ali and M. Shah. Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 32, (2010) 288–303.

[5] M. Ahmad and S.-W. Lee. Human action recognition using shape and CLG-motion flow from multi-view image sequences. *Pattern Recognition* 41, (2008) 2237–2252.

[6] J. Liu and M. Shah. Learning human actions via information maximization. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on. 2008 1–8.

[7] S. Danafar, A. Giusti, and J. Schmidhuber. Novel Kernel-Based Recognizers of Human Actions. *EURASIP J. Adv. Sig. Proc.* 2010.

[8] A. Yilmaz and M. Shah. Actions sketch: a novel action representation. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1. 2005 984–989 vol. 1.

[9] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011 3337–3344.

[10] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010 2046–2053.

[11] X. Wu, D. Xu, L. Duan, and J. Luo. Action recognition using context and appearance distribution features. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011 489–496.

[12] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In Computer Vision–ECCV 2012, 425–438. Springer, 2012.

[13] S. Sadanand and J. Corso. Action bank: A high-level representation of activity in video. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. 2012 1234–1241.

[14] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. IEEE, 2011 3169–3176.

[15] A. Dileep and C. Chandra Sekhar. Class-specific GMM based intermediate matching kernel for classification of varying length patterns of long duration speech using support vector machines. *Speech Communication* 57, (2014) 126–143.

[16] N. Smith and M. Niranjan. Data-Dependent Kernels In Svm Classification Of Speech Patterns 2001.

[17] K.-A. Lee, C. You, H. Li, and T. Kinnunen. A GMM-based probabilistic sequence kernel for speaker verification. In INTERSPEECH. Citeseer, 2007 294–297.

[18] W. M. Campbell, D. E. Sturim, and D. A. Reynolds. Support vector machines using GMM supervectors for speaker verification. *Signal Processing Letters, IEEE* 13, (2006) 308–311.

[19] S. Boughorbel, J.-P. Tarel, and F. Fleuret. Non-Mercer Kernels for SVM Object Recognition. In In British Machine Vision Conference (BMVC. 2004 137–146.

[20] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on. IEEE, 2003 257–264.

[21] S. Boughorbel, J. P. Tarel, and N. Boujemaa. The intermediate matching kernel for image local features. In Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on, volume 2. IEEE, 2005 889–894.

[22] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77, (1989) 257–286.

[23] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of computational biology* 7, (2000) 95–114.

[24] N. Smith and M. Gales. Speech Recognition using SVMs. In Advances in Neural Information Processing Systems 14. MIT Press, 2002 1197–1204.

[25] T. Jebara, R. Kondor, A. Howard, K. Bennett, and N. Cesa-bianchi. Probability product kernels. *Journal of Machine Learning Research* 5, (2004) 819–844.

[26] A. Dileep and C. Sekhar. HMM Based Intermediate Matching Kernel for Classification of Sequential Patterns of Speech Using Support Vector Machines. *Audio, Speech, and Language Processing, IEEE Transactions on* 21, (2013) 2570–2582.

[27] N. Dalal and B. Triggs. Histograms of Oriented Gradients for Human Detection. In C. Schmid, S. Soatto, and C. Tomasi, eds., International Conference on Computer Vision & Pattern Recognition, volume 2. INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, 2005 886–893.

[28] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos x201C;in the wild x201D;. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. 2009 1996–2003.

[29] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402* .

[30] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In Proceedings of the International Conference on Computer Vision (ICCV). 2011 .