FEATURE PRUNING FOR ACTION RECOGNITION IN COMPLEX ENVIRONMENT

by

ADARSH NAGARAJA
B.E. Electronics and Communication, 2006

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2011

Major Professor: Marshall Tappen

ABSTRACT

A significant number of action recognition research efforts use spatio-temporal interest point detectors for feature extraction. Although the extracted features provide useful information for recognizing actions, a significant number of them contain irrelevant motion and background clutter. In many cases, the extracted features are included as is in the classification pipeline, and sophisticated noise removal techniques are subsequently used to alleviate their effect on classification. We introduce a new action database, created from the Weizmann database, that reveals a significant weakness in systems based on popular cuboid descriptors. Experiments show that introducing complex backgrounds, stationary or dynamic, into the video causes a significant degradation in recognition performance. Moreover, this degradation cannot be fixed by fine-tuning the system or selecting better interest points. Instead, we show that the problem lies at the descriptor level and must be addressed by modifying descriptors.

To my parents and for all the inspirations

Acknowledgments

I sincerely express my greatest grattitude to my advisor, Assistant Professor Dr Marshall Tappen for all the elaborate guidance, support and encouragement without which this work would have been impossible. I have learnt and tried to emulate his approach of looking at the problem and solving it.

I would like to thank Syed Zain Masood for all the discussion and collabration of the work which played a significant role in my thesis.

I would like to thank my thesis committe members, Professor Dr Hassan Foroosh and Associate Professor Dr Niels da Vitoria Lobo.

I would also like to thank my lab mates such as Nazar Khan, Chris Ellis, Paul Scovanner and Zhonkai Han for all the discussions and memories.

TABLE OF CONTENTS

LIST OF FIGURES ix

LIST OF TABLES xiii

CHAPTER 1: INTRODUCTION 1

1.1 Problem definition 2

1.2 contribution 3

CHAPTER 2: BACKGROUND 4

2.1 Bag of Words Model based Action Recognition 4

2.2 Dollar's Feature extraction method 8

2.3 Laptev's feature extraction method 11

2.4 Results of bag of words model 12

CHAPTER 3: DATA SETS 14

3.1 KTH 14

3.2 WEIZMANN 16

3.3 UCF Sports 17

3.4 Youtube 18

3.5 Synthesized dataset 19

3.5.1 Need for synthetic dataset 19

3.5.2 Construction methods 20

3.5.3 Addressing Matting Artifacts 22

3.5.4 Construction choices 22

3.5.5 Results on Synthesized dataset 23

CHAPTER 4: FEATURE PRUNING 24

4.1 Parameter Fine tuning 24

4.1.1 Temporal Scale 25

4.1.2 Clustering and Histogram 25

4.2 Descriptor Pruning 26

4.3 Localization 28

4.3.1 Automatic Localization 30

4.4 Interest point pruning 31

4.5 Cuboid Masking 33

4.6 UCF Sports 37

CHAPTER 5: DISCUSSION AND CONCLUSIONS 39

LIST OF REFERENCES 40

LIST OF FIGURES

2.1 The Diagram shows the Bag of Words model applied for object recognition.The first Row shows the set of images. The second row shows the set of key zones in the images which are represented using image features. The third row shows the bag of visual words formed from clustering the key zones. The fourth row shows the histogram of visual words 7

2.2 The Diagram shows the video from which cuboids are selected from key interest points. The concatenated pixel values of consecutive images in the cuboid volume are also shown. The descriptors are calculated from this strip of images to represent the cuboid 10

2.3 The Diagram shows the HOG Descriptors.(a)Full Descriptors with 2x2x2 histogram cells (b)Histogram computation over 2x2x2 cell (c)gradient orientation quantization (d)mean gradient computation 10

3.1 The figure shows the 6 different actions of KTH dataset 15

3.2 The confusion matrix shows the accuracy achieved by our Feature pruning method 4 on KTH dataset. It can be observed that most of misclassification is happening between jogging and running actions. 15

3.3 The figure shows the 10 different actions of Weizmann dataset 16

3.4 The figure shows the 11 different actions of Weizmann dataset 17

3.5 The figure shows the 11 different actions of Weizmann dataset 18

3.6 Examples of the Weizmann original (top row) and Weizmann static complex (bottom row) datasets. Each video in the static dataset has the exact same complex background image for the entire sequence. This indicates the background complexity of gradients, textures and contrasts on which the actions are overlayed. We reiterate that background remains same throughout a static complex video sequence and since we are not using any human detector approach, presence of humans in the background does not confuse our system . 21

3.7 Examples of theWeizmann dynamic dataset. The figure shows the 1, 11, 21, 31 and 41 frames of 2 running actions with complex dynamic background. The top row indicates running action overlayed on the fast moving trees video with high gradients and textures. Bottom row indicates running action overlayed on the slow moving eagle video 21

4.1 Flattened-in-time display of frames of cuboids for boxing (left table) and running (right table) actions from the KTH database. Each table shows action-relevant cuboids (2nd column), irrelevant human motion cuboids (3rd column) and background motion cuboids (4th column). The last two rows, respectively contain, average values of the magnitude of optical flow vectors in the cuboids and of the response of Dollars temporal Gabor filter [4] at the cuboid centers, for each class of cuboids. Relevant cuboids have higher values for both measures and thus can be distinguishable from irrelevant cuboids using simple adaptive thresholding. 27

4.2 Top row shows the intereste points without pruning for Weizman simple, static and Dynamic. Bottom row shows the interest points for the same frame after pruning. 31

4.3 The figure shows the effect of cuboid masking. **Column 1:** Shows the same running action performed by the same person matted on 3 different complex moving backgrounds. **Column 2:** Shows cuboids extracted from each video sequence. Size of each cuboid is 13x13x7, where all 7 frames are shown in a single row. **Column 3:** Illustrates the exact same cuboids as in column 2 after applying cuboid masking. **Column 4:** Shows the temporal gradients of cuboids in column 2. **Column 5:** Shows the temporal gradients of cuboids in column 3. The gradient in column 4 corresponding to background content (red outlined) appear different for each video sequence. However, the gradients of all three actions looks similar after applying cuboid masking, as depicted in column 5. This is confirmed by average SSIM values of 0.67 and 0.75 for original temporal gradients (column 4) and cuboid masked temporal gradients (column 5) respectively. 34

LIST OF TABLES

2.1 Comparison of our baseline and other state-of-the-art techniques on well known datasets. 13

3.1 This table shows results obtained by using simple baseline methods. We see that recognizing the same simple actions in presence of complex backgrounds (static or dynamic) significantly affects performance. Accuracy on dynamic is worse due to the presence of background motion in the video sequences. . . 23

4.1 Different Temporal Scales 25

4.2 We experimented with different cluster sizes for Weizmann dataset. It can be seen that tweaking the cluster size parameter does not solve the problem. . . 26

4.3 Effect of pruning. Recognition percentages for classification on KTH dataset using histograms obtained from non-pruned interest points, using histograms obtained after pruning based on magnitude of optical flow vectors, using histograms obtained after pruning based on Gabor filter response and by concatenating histograms obtained from both methods of pruning 29

4.4 The above table shows the accuracy on synthesized complex dataset when using interest point pruning with automatic localization. Best possible results for interest point pruning with ground-truth localization are also shown. Although results improve, they are still not comparable to those achieved on original Weizmann dataset using our baseline system (Table 2.1). 33

4.5 The above table shows the accuracy on Weizmann dynamic dataset using combination of Interest Point Pruning (IPP) and Cuboid Masking (CM) w.r.t **Automatic masks**. We can see that optimal accuracy is achieved when using both IPP and CM strategies. 35

4.6 The above table shows the accuracy on Weizmann dynamic dataset using combination of Interest Point Pruning (IPP) and Cuboid Masking (CM) w.r.t **Ground truth masks**. We can see that optimal accuracy is achieved when using both IPP and CM strategies. 37

4.7 The table shows the results on UCF sports with **Automatic mask**. It is evident that IPP and CM strategies improve the accuracy by 12% 38

4.8 The table shows the results on UCF sports with **Ground-truth mask**. It is evident that IPP and CM strategies improve the accuracy by 17% 38

CHAPTER 1: INTRODUCTION

Over the past decade, the cameras have become ubiquitous and this has led to the creation of large amount of video data. The extensive growth of video content can be managed and analysed only with the understanding of the video. Though human vision systems perform this job in a very simple and easy way, it is a challenging and open ended problem for a computer. The computers needs to understand the events, actions and activities happening in the video to manage and tag the video with the suitable label. To understand and address this problem, building an action recognition system is of prime importance. The action recognition system will help in understanding the video which involves action/activities performed by humans.

Human action recognition is the phenomenon of recognizing the actions performed by humans in the video. Action recognition is one of the challenging problem in computer vision with the application ranging from content indexing, video gaming, animation to video surveillance. Initial action recognition systems are learnt and tested in controlled environments, which involved a single person performing a simple action such as jogging, waving etc on a simple background and captured using a stationary camera. More advanced recognition algorithms were proposed for complex scenes which included the presence of occlusion,

scale adjustments, intra-class variations, background clutter, changes in illumination and attributes of individuals performing the actions.

1.1 Problem definition

The bag of words based action recognition systems achieve near 100% accuracy for simple data sets such as KTH and Weizmann. The extension of same method to relatively complex datasets leads to only sub-optimal performance of the recognition system. The complex datasets are captured in less controlled environments such as sport clips, movie clips and home made videos which are realistic videos. Simple data sets have coherent actions performed by single actor captured from a fixed camera. The only variations present in the datasets were the difference in clothing and the way of performing the actions by different actors. However, the realistic videos had many parameters which were varying from videos to videos. For examples, cameras which were capturing the video had movements. This could imply, even though the person is cycling, his/her real world movement might get compensated with camera movement and the person's location might be stationary with respect to video. The videos might have illumination change and background clutter. These variations will influence as key zones in the video and could mislead the system. The simpler datasets had very distinct and unique set of actions to classify such as waving and walking which are very different. But, realisitic videos have inter class similarities. For example, videos of kicking the ball have running actions also, which might confuse the recognition system.

1.2 contribution

The complex realistic dataset poses more challenging problems and mere extension of simple bag of words model will only give sub-optimal accuracy. In this work, we present a detailed study of the problems associated with complex datasets. We create a synthetic dataset with background complexity and also propose the solutions to improve the performance of the recognition with the systems built on bag of words model. We show the cuboid pruning and masking techniques improve the recognition accuracy.

CHAPTER 2: BACKGROUND

In this section, we explain the bag of words based action recognition model. We will explain two most popular feature extraction methodologies i.e. Dollar's and Laptev's methods. We will also explain the SVM based classifier.

2.1 Bag of Words Model based Action Recognition

The bag of words based recognition system was popular approach for document analysis. The same bag of words approach model can be extended to object and action recognition. The words represent the visual information rather than the textual information. In case of action recognition, features representing the patch of video represent the visual word.

The bag of words model is used as supervised learning model. The Leave one out cross validation(LOOCV) is followed to obtain the final accuracy. The dataset is divided into mutually exclusive set of training and testing videos. The training videos are considered for building the vocabulary of visual words. The testing videos are matched with vocabulary to build histogram.

For a given training video

- Detect Space time interest points:

  These interest points are the zones of interest in visual sense. These interest points

represent regions in video with high spatial gradient, texture, temporal gradients or optical flow.

- Calculate Descriptors for interest points:

  The descriptor is calculated for the small region surrounding each of the interest point. The descriptor may represent the histogram of gradients or histogram of optical flow.

- Form Vocabulary:

  The descriptors of all the training videos are considered and K-means clustering is applied on the video. The number of cluster centers is pre determined based on the dataset and number of action classes. The cluster center represent the visual words of the vocabulary. K-means clustering is given by following equation.

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} (||x_j - \mu||^2) \tag{2.1}$$

  where k is the number of clusters, x_j is the cuboid descriptor. Minimizing this gives the set of visual words $S = [S_1, S_2, ..S_k]$. By default, vocabulary size of 500 clusters is used.

- Histogram of videos:

  All the training videos have descriptors and these descriptors are matched to one of the visual words based on L2-norm. Histograms are built for each video with number of bins being equal to number of cluster centers. Based on the matching, each bin is incremented and thus a histogram is formed for each video in the training set.

- SVM classifier:

  The Support vector machine is the supervised learning algorithm which fits the hyper plane in feature space to divide the data. It is one of the most advanced classifier and most suitable for high dimensional feature space represented by histograms. SVM classifiers have training and testing phase. In training phase, histograms of the training video and the labels of the histograms are fed to learn support vectors for multi-class classification problem. In testing phase, labels of the testing histograms are found from histograms of testing videos and the SVM parameters learnt during training.

  The LibSVM package is used for SVM training and testing.

$$\min(\sum_{i=1}^{k}(x - S_i))^2 \tag{2.2}$$

  gives the affiliation of cuboid descriptor x to cluster S_i and makes a unary increment at index S_i in the histogram. $H = h(S_1), h(S_2)..h(S_k)$ is formed for each video where $h(S_i)$ for i 1 to k, is the number of voting for the visual word S_i. We experimented with Histogram intersection kernel and χ^2 kernel as kernel functions for SVM. The histogram intersection kernel is given by

$$K(\alpha, \beta) = \sum_{i=1}^{k}(\min(\alpha_i, \beta_i)) \tag{2.3}$$

  where α and β are the histograms with k bins.

  The χ^2 kernel is given by

$$K(\alpha, \beta) = 1 - \sum_{i=1}^{k}\frac{(\alpha_i - \beta_i)^2}{(\alpha_i + \beta_i)} \tag{2.4}$$

Figure 2.1: The Diagram shows the Bag of Words model applied for object recognition.The first Row shows the set of images. The second row shows the set of key zones in the images which are represented using image features. The third row shows the bag of visual words formed from clustering the key zones. The fourth row shows the histogram of visual words

where α and β are the histograms with k bins.

The above paragraphs explained the series of procedure followed in learning stage of the bag of words based action recognition model. The outcome of the learning phase are the visual words representing the vocabulary and SVM parameters for classification. In training stage, the labels of the histograms of the videos are known, however in testing action labels of the videos should be determined. Following steps are followed for any given video of unknown action label.

For a given testing video

- Detect Space time interest points
- Calculate Descriptors for interest points
- Form Histogram by comparing descriptors of the video with visual vocabulary
- Classify the histogram using SVM to get the label of the video

2.2 Dollar's Feature extraction method

Bag of words based action recognition model relies on interest point detection and feature descriptor calculation. Dollar *et al.* proposed [4] the interest point detector and feature descriptor using gradients and optical flow features. Following steps describe the Dollar's method for feature extraction Given any video sequence,

- Detect spatio-temporal interest points:

  The interest points are defined as local maxima of the response function, using Dollar *et al.* [4] code provided on-line[1].

$$R = (I * g * h_{ev}) + (I * g * h_{od}) \tag{2.5}$$

  where $g(x, y; \sigma)$ is a Gaussian smoothness kernel across the spatial domain and

$$h_{ev}(t; \tau, \omega) = -cos(2\pi t\omega)e^{-t^2/\tau^2}$$

$$h_{od}(t; \tau, \omega) = -sin(2\pi t\omega)e^{-t^2/\tau^2}$$

[1] http://vision.ucsd.edu/ pdollar/toolbox/doc/

are quadrature pair of 1D Gabor filters applied temporally. Once detected, cuboids are extracted centered around the interest points. The size of cuboids is decided based on the size of the video. For the videos of resolution less than 640x480, 13x13 spatial are used as cuboid size. Temporal cuboid size are varied from 5,7,9 to 11 for different results. The size of cuboid determines the local patch of video which describes the video locally and variations in size of cuboid results in variation in final recognition accuracy.

- Compute descriptors:

  For a given point (x, y, t, σ, τ) the gradients are calculated for each cuboid centering at (x, y, t) with

$$\Delta_x(\sigma) = 2 \times ceil(3\sigma) + 1$$

$$\Delta_y(\sigma) = 2 \times ceil(3\sigma) + 1$$

$$\Delta_t(\tau) = 2 \times ceil(3\tau) + 1$$

  as the spatial and temporal scales respectively (using Dollar *et al.* [4] code provided on-line[1]). These histogram of gradients (HoG) are concatenated to form a high dimensional vector. Gradients in the video represent the edges in the images and accumulation of these edges over time give the movement of edges. Histograms can be built on 2D or 3D. Optical Flow or pixel values can also be used as features instead of gradients. Principal Component Analysis (PCA) is applied to project the high dimensional feature vector into lower dimensional space of 100 dimensions. The lower dimensional

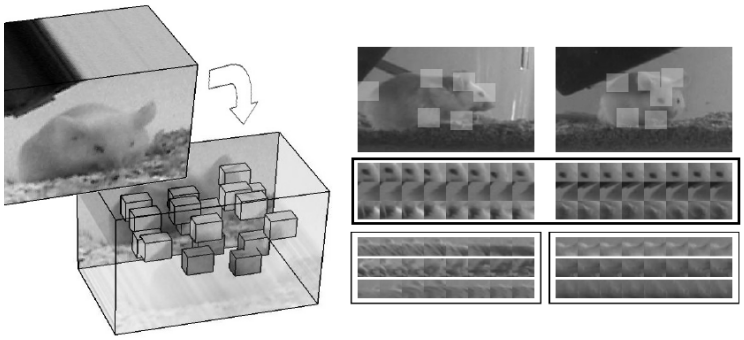

Figure 2.2: The Diagram shows the video from which cuboids are selected from key interest points. The concatenated pixel values of consecutive images in the cuboid volume are also shown. The descriptors are calculated from this strip of images to represent the cuboid

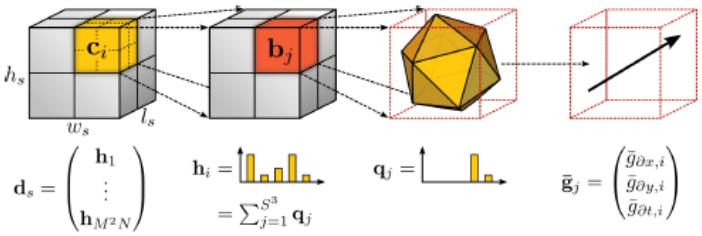

Figure 2.3: The Diagram shows the HOG Descriptors.(a)Full Descriptors with 2x2x2 histogram cells (b)Histogram computation over 2x2x2 cell (c)gradient orientation quantization (d)mean gradient computation

representation helps making a concise representation and faster manipulation of descriptor data.

2.3 Laptev's feature extraction method

Bag of words based action recognition model relies on interest point detection and feature descriptor calculation. Laptev *et al.* proposed [2] the interest point detector and feature descriptor using gradients and optical flow features. Following steps describe the Laptev's method for feature extraction

Given any video sequence,

- Detect spatio-temporal Harris corners:

  The 2D Harris corners were extended to space-time domain. The local positive maxima's of H are considered as space time interest points, where

$$H = det(\mu) - trace^3(\mu) \tag{2.6}$$

  where

$$(\mu) = g(x, y, t, \sigma, \tau) * \begin{pmatrix} L_x^2 & L_x L_y & L_x L_t \\ L_x L_y & L_y^2 & L_y L_t \\ L_x L_t & L_y L_t & L_t^2 \end{pmatrix}$$

$$g(x, y, t, \sigma, \tau) = \frac{1}{\sqrt{(2\pi)^3 \sigma^4 \tau^3}} . exp(-\frac{x^2 + y^2}{2\sigma^2} - \frac{t^2}{2\tau^2}) \tag{2.7}$$

  and L_x, L_y and L_t are the first order derivatives in x, y and t direction

- Calculate HOG/HOF descriptors:

  For each interest point, the Histogram of oriented gradients or Histogram of optical flow is calcualated for cells of size MxMxN. The mean of the gradient or optical flow

is calculated and it is quantized to one of the 4 to 5 bins to form histograms and the histograms of all the cells are concatenated. For better accuracy, histogram of oriented gradients are concatenated with histogram of optical flow to form descriptor.

2.4 Results of bag of words model

The above sections explained the bag of words model and 2 popular extraction methodologies needed for bag of words. The above mentioned methods are followed for building action recognition system for data sets such as Weizmann, KTH, Youtube and Synthetic Weizmann(Static and Dynamic Weizmann). The datasets are analysed in chapter 3. The Table 2.1 gives the recognition accuracy obtained from leave one out cross validation. From the table it is evident that simple data sets such as Original Weizmann and KTH achieve near 100% accuracy where as complex and realistic datasets such as Youtube and synthesized Weizmann achieve not more than 73.5%. The table also shows the minor difference in accuracy between Dollar and Laptev's feature extraction method.

Dataset	Dollar	Laptev (HOF) [13]	Laptev(HOG3D) [12]
Original Weizmann	98%	92%	91%
Static Weizmann	73.5%	76%	61.5%
Dynamic Weizmann	36.5%	31%	46%
KTH	93.5%	92%	90%
Youtube	65%	–	–

Table 2.1: Comparison of our baseline and other state-of-the-art techniques on well known datasets.

CHAPTER 3: DATA SETS

To solve the problem of Action recognition,different datasets were gathered from different research communities. Each of the datasets poses the recognition problem with different difficulties and it is important to analyze the datasets.

3.1 KTH

KTH contains 6 different types of actions (Figure 3.1): boxing, hand clapping, hand waving, jogging, running and walking. These actions are performed by 25 different people in 4 different scenarios (indoor, outdoor, clothing change and scale variations) making it a total of 598 videos.

Recent action recognition systems have near 100% accuracy (figure 3.2) on all actions except jogging and running [8, 18, 16, 13]. This is because the difference between these actions is not discernible for portions of this dataset, such as the videos from person 2. To justify this decision, we conducted an experiment, involving humans, to gauge the difficulty of correctly recognizing actions between jogging and running. Each person was shown 2 training videos of each jogging and running and then was asked to correctly label a total of 50 test videos. We found the surprising result that human subjects were only able to correctly recognize 90% of the videos shown. This is approximately the same accuracy that

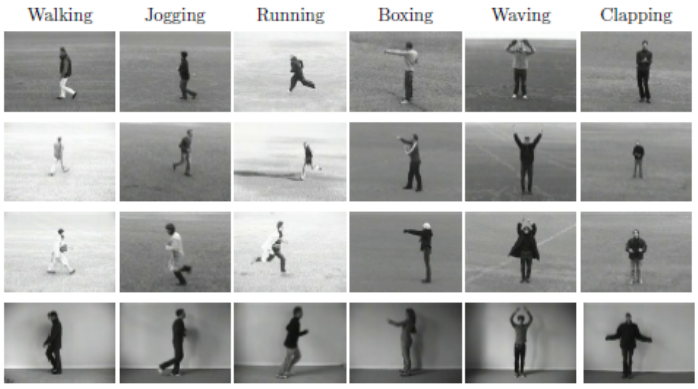

Figure 3.1: The figure shows the 6 different actions of KTH dataset

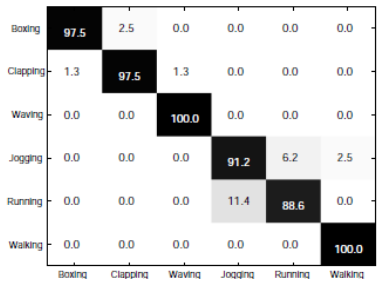

Figure 3.2: The confusion matrix shows the accuracy achieved by our Feature pruning method 4 on KTH dataset. It can be observed that most of misclassification is happening between jogging and running actions.

state of-the-art systems achieve. The difficulty that humans have with this set makes it less desirable for evaluating machine vision systems

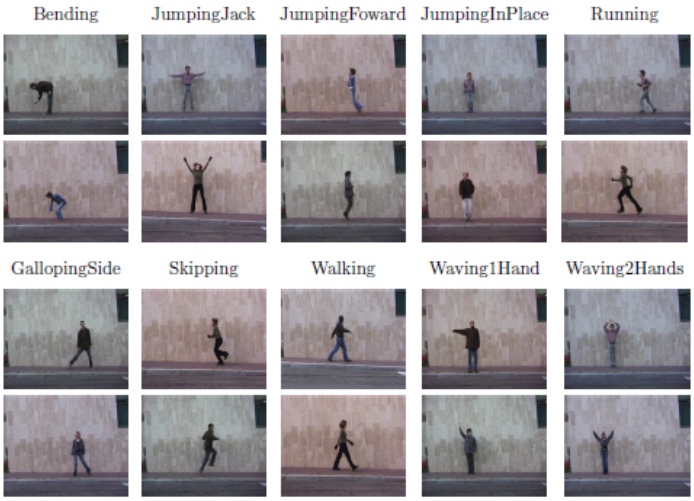

Figure 3.3: The figure shows the 10 different actions of Weizmann dataset

3.2 WEIZMANN

The Weizmann actions dataset (figure 3.3) [1] consists of ten different types of action classes: bending downwards, running, walking, skipping, jumping-jack, jumping forward, jumping in place, galloping sideways, waving with two hands, and waving with one hand. Each action class is performed once (sometimes twice) by 9 subjects resulting in 93 video sequences in total. The background in the videos is homogeneous and static. We achieve 98% accuracy on Weizmann dataset.

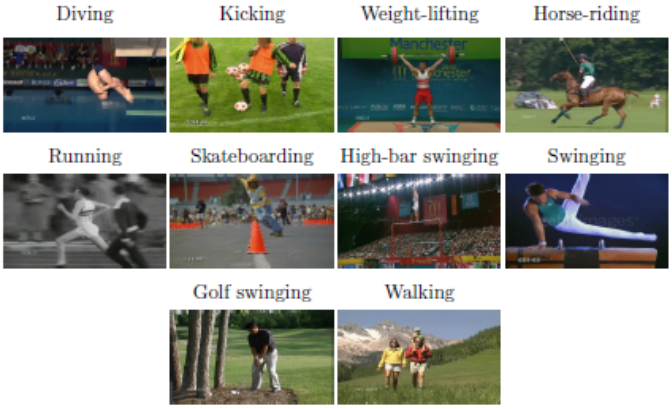

Figure 3.4: The figure shows the 11 different actions of Weizmann dataset

3.3 UCF Sports

The UCF sport actions dataset [Rodriguez et al., 2008] contains ten different types of human actions: swinging (on the pommel horse and on the floor), diving, kicking (a ball), weight-lifting, horse-riding, running, skateboarding, swinging (at the high bar), golf swinging and walking (figure3.4). The dataset consists of 150 video samples which show a large intra-class variability. The performance criterion for the multi-class task is the average accuracy over all classes. We achieve 70% accuracy with simple bag of words and we improve it to 85% accuracy UCF Sports dataset using the method explained in chapter 4.

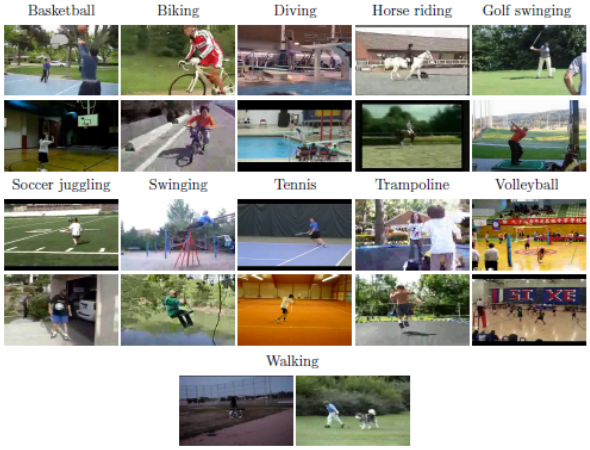

Figure 3.5: The figure shows the 11 different actions of Weizmann dataset

3.4 Youtube

The YouTube dataset has been introduced by Liu and 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 3.5. This dataset is challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background,illumination conditions etc. The dataset contains a total of 1600 sequences. In the original setting, the evaluation is carried out using cross validation for a set of 25 folds that is defined by the authors. Average accuracy over all classes is used as performance measure. We achieve 71.5% accuracy using feature pruning explained in section4.

3.5 Synthesized dataset

3.5.1 Need for synthetic dataset

KTH and Weizmann datasets contain actors performing simple periodic actions with simple fixed backgrounds. This construction forces the recognition system to focus on directly recognizing the action being performed by the actor. Also, both these datasets present additional complexity over the simple datasets. This is useful in isolating the negative role that different complexities play in recognizing actions and thus provide a thorough analysis of the problem. The realistic datasets such as UCF Sports and Youtube achieve accuracies not more than 75%. The reason for lower accuracies for these datasets is the quantum jump in complexity which makes it harder to analyze. Youtube dataset has occlusion, multiple people, scene changes, camera motion, complex background and inter class overlaps.

In order to address the action recognition problem in complex environments, it is desirable to have datasets with incremental complexities rather than the quantum leap in complexities posed by realistic action datasets such as Youtube. Hence, we create a synthetic dataset using the action masks of Weizmann dataset and use complex background. The newly created dataset contains simple actions performed by single actor, but in a complex static or dynamic background. The reasoning behind this data set is that the central recognition problem remains the same, but the task is made more difficult by the addition of the complex background.

3.5.2 Construction methods

We create two new datasets using Weizmann action masks and background from Youtube videos. We downloaded a total of 15 Youtube videos making sure that each of them contain some complex scene. We then randomly select a Youtube video from this pool and perform matting with one of the Weizmann action mask. Keeping the Youtube video pool considerably lower than the number of action masks (93 in this case) ensures different actions being performed on the same background and thus diminishing the role of background in differentiating actions. Two different data-sets are developed using this strategy:

- Weizmann Static: For this dataset, instead of choosing the whole Youtube video as a background, we pick a single frame and perform matting to form the new action video (figure 3.6). The name static is because the background, although complex, is fixed. This helps us analyze how a static complex background affects recognition.

- Weizmann Dynamic: For this, the whole video is matted with the action masks, refer figure 3.7. The moving background makes it a much harder problem to recognize actions.This helps to analyze how the camera motion on a complex background affects the recognition

It should be noted that when creating the dynamic set,we make sure that none of the Youtube backgrounds have humans in it. This is a necessity as the presence of humans in Youtube background videos is most likely to be accompanied by some action, leading to multiple actions in a single video. Since the static case is only a single frame, no such

Figure 3.6: Examples of the Weizmann original (top row) and Weizmann static complex (bottom row) datasets. Each video in the static dataset has the exact same complex background image for the entire sequence. This indicates the background complexity of gradients, textures and contrasts on which the actions are overlayed. We reiterate that background remains same throughout a static complex video sequence and since we are not using any human detector approach, presence of humans in the background does not confuse our system

Figure 3.7: Examples of theWeizmann dynamic dataset. The figure shows the 1, 11, 21, 31 and 41 frames of 2 running actions with complex dynamic background. The top row indicates running action overlayed on the fast moving trees video with high gradients and textures. Bottom row indicates running action overlayed on the slow moving eagle video

restriction is applied to it. Our methodology of creating a complex dataset for simple actions is different from [15]. Our synthesized datasets are complete replicas of the simple dataset in terms of the action being performed, and accuracy of the recognition can be compared directly. Since, we use matting [14] to create new datasets, it wont add any biases, due to change in the actor performing the action. Because of the synthetic construction of these datasets, matting artifacts could pose an issue.

3.5.3 Addressing Matting Artifacts

To measure this, we constructed a third dataset by matting the Weizmann action masks with a simple static gray background. Testing results for this dataset were similar to what we achieved on the original Weizmann dataset. This indicates that matting does not cause a performance drop. We use ground truth silhouette mask provided in Weizmann dataset only for synthetic dataset formation.

3.5.4 Construction choices

The Weizmann dataset was chosen because the actions are simple and coherent. In addition, each video has an associated action mask which makes it possible to extract the action and construct new videos with complex backgrounds. KTH dataset though is a simple dataset, the action masks are not available, but most importantly, as explained in section 3.1, difference between jogging and running actions are not discernible for humans with 100% accuracy. We avoid the use of realistic complex datasets like Youtube [18, 16] and Hollywood

Dataset	Original	Static Weizmann	Dynamic Weizmann
Accuracy	98%	73.5%	36.5%

Table 3.1: This table shows results obtained by using simple baseline methods. We see that recognizing the same simple actions in presence of complex backgrounds (static or dynamic) significantly affects performance. Accuracy on dynamic is worse due to the presence of background motion in the video sequences.

[13, 19] because isolating the effect of background complexity from within the highly complex structure (multiple people, multiple actions, camera movement, high diversity within action class) of these datasets is extremely challenging. This makes Weizmann dataset the obvious choice for construction.

3.5.5 Results on Synthesized dataset

The table 3.1 shows the results obtained on synthetic Weizmann dataset using Bag of words model. Even though the dataset contains simple actions performed by single actor with no camera motion and occlusion, accuracy of the recognition goes down from 98% for original Weizmann to 73.5% for static. The presence of gradient in the background reduces the accuracy by 24.5%. The accuracy is even worser for Dynamic dataset as it reduces to 36.5% due to the motion in background. These results clearly indicates the fragility of Dollar's descriptor and STIPS's descriptor based bag of words model. In the next chapter, feature pruning method is explained which improves the accuracy of the recognition on synthetic dataset.

CHAPTER 4: FEATURE PRUNING

Previous chapters (Chapter 2) described the bag of Words model and the feature point selection and descriptor calculation methods. These methods give very high accuracy for simple datasets such as KTH and Weizmann. But the accuracy decreases for realistic dataset such as UCF Sports, Youtube. We created synthetic wiezmann dataset by adding background complexity to the Weizmann dataset (Section3.5) and found the large decrease in accuracy. The synthesized dataset helps in analysing the role of background complexity and poses the problem of recognition with incremental complexity unlike Youtube and UCF Sports.

Some of the researcher's claim that parameter fine tuning can improve recognition and hence we show the results from parameter fine tuning. The proposed Feature pruning techniques such as interest point pruning, cuboid masking and automatic mask generation described in the section following parameter fine tuning.

4.1 Parameter Fine tuning

The feature descriptor based Bag of Words model for action recognition has many parameters which are selected based on the dataset and these parameters when changed, influences the recognition accuracy. We experimented with different fine tunable parameters and found out that the parameter fine tuning has minor improvement in the accuracy.

Temporal Scale	Simple	Static	Dynamic
5 frames	92.5%	69%	23%
7 frames	98%	73.5%	36.5%
9 frames	93%	67.5%	46.0%
11 frames	88%	63.5%	37%
Concatenated	95.5%	77.5%	34%

Table 4.1: Different Temporal Scales

4.1.1 Temporal Scale

The cuboids temporal size decides the number of frames considered within each cuboid. Laptev *et al.* [13] suggest using multiple spatio-temporal scales for improved performance while [7] show that averaging across all available features performs remarkably well. For this purpose, we experimented by calculating results for temporal sizes of 5,7,9,11 frames and average of all temporal scales. We see that some temporal scales perform better than others and the averaging is an improvement over the low accuracy configurations.

4.1.2 Clustering and Histogram

In the process of making the vocabulary, the cuboid descriptors of all actions are accumulated and clustered, where each cluster center represents a visual word. It can be argued that picking the right number of clusters can have a significant impact on the recognition accuracy. However, we observed that for our current problem that is not the case. To show this, we experimented with different cluster sizes and present the results in Table 4.2. We see slight

Dataset	250	500	1000
Simple Weizmann	95.5%	98%	98%
Static Weizmann	71.5%	73.5%	74.5%
Dynamic Weizmann	34.5%	36.5%	34%

Table 4.2: We experimented with different cluster sizes for Weizmann dataset. It can be seen that tweaking the cluster size parameter does not solve the problem.

variations in results but nothing significant enough to suggest that tuning of the cluster number parameter is solution to the posed problem.

4.2 Descriptor Pruning

It is observed that a significant number of cuboids extracted using Dollars feature extractor [4] are detected due to background clutter or highly textured foreground areas and irrelevant human motion. Since these cuboids have no discriminating connection with the action being performed, it is best to remove them. Figure 4.1 shows some of the cuboids extracted from a boxing and a running video in the KTH database. One can observe that the first grouping of cuboids clearly depicts the action taking place. In contrast, the second and third groupings for each action are merely a result of non-crucial human motion and background noise respectively. These cuboids exhibit information irrelevant to the action being performed. Therefore, it is important that we prune these features as early as possible to prevent them from causing problems later.

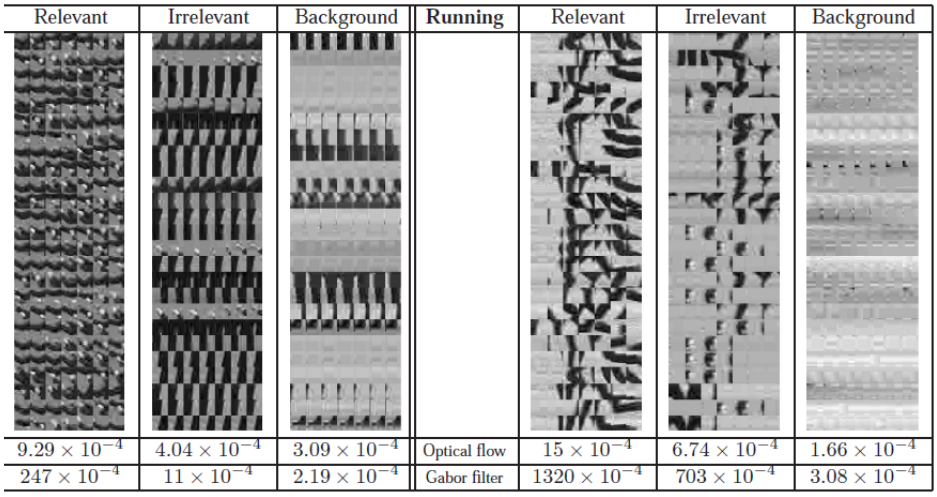

Figure 4.1: Flattened-in-time display of frames of cuboids for boxing (left table) and running (right table) actions from the KTH database. Each table shows action-relevant cuboids (2nd column), irrelevant human motion cuboids (3rd column) and background motion cuboids (4th column). The last two rows, respectively contain, average values of the magnitude of optical flow vectors in the cuboids and of the response of Dollars temporal Gabor filter [4] at the cuboid centers, for each class of cuboids. Relevant cuboids have higher values for both measures and thus can be distinguishable from irrelevant cuboids using simple adaptive thresholding.

Since action-relevant cuboids capture most of the descriptive action being performed in the video, they exhibit relatively high motion content. Cuboids with low motion content are most probably a result of background noise, highly textured foreground areas or irrelevant human motion. Action-relevant cuboids can therefore be differentiated from irrelevant cuboids based on their motion content. For this purpose, we use two motion criterion:

- The response of the temporal Gabor filter from [4] at the interest point.

- The cumulative magnitude of the optical flow vectors in the cuboid surrounding the interest point defined as

$$\mu = \sum_{v} (\sqrt{v_x^2 + v_y^2}) \tag{4.1}$$

where v is a voxel in the cuboid and v_x and v_y are the horizontal and vertical optical flow components.

Average values for both motion measures are given under the corresponding cuboid groupings in Figure 4.1. Motion content averages for action-relevant cuboids are higher than those for action-irrelevant cuboids. A simple thresholding of these two measures is used to prune irrelevant cuboids. The formula used for computing such an adaptive threshold, α, for a video is $\alpha = max - \lambda(max - min)$ where min and max are the minimum and maximum values of the motion measure for all cuboids in the video and $lambda \in [0, 1]$ is a constant that determines the final threshold value.

The descriptor pruning method improved accuracy of Youtube dataset from 65%(achieved from baseline bag of words method) to 71.5%. However the table 4.3 shows the marginal improvement in results for KTH dataset.

4.3 Localization

We observed that the introduction of complex background in videos of simple actions greatly affects recognition performance (refer to section 3.5.5). Since the only change between the

	Accuracy(%)
No Pruning	93.9%
Optical Flow magnitude	94.1%
Gabor Filter response	94.7%
Concatenating both	95.8%

Table 4.3: Effect of pruning. Recognition percentages for classification on KTH dataset using histograms obtained from non-pruned interest points, using histograms obtained after pruning based on magnitude of optical flow vectors, using histograms obtained after pruning based on Gabor filter response and by concatenating histograms obtained from both methods of pruning

original and dynamic datasets is of the background, it is reasonable to say that the drop in accuracy is only due to the change in background complexity. This is because increased background complexity leads to detection of irrelevant background interest points that are a main source of performance degradation. One would assume that eliminating these background interest points should solve the problem. However, *that is not the case*. In fact, it the use of localization for both pruning irrelevant interest points and eradicating background corruption inside cuboids that leads to optimal results. Thus we can say that:

- Action localization is important but
- Application/use of localization is equally significant

We propose a stepwise solution to the above posed problem:

- First and foremost, we need a good automatic action localization methodology (preferably a tight bounding box around the person performing the action).
- Once we have localization information, we eliminate all interest points detected due to background motion
- Having removed erroneous interest points, we use localization to correct cuboid corruption due to background information i.e. mask out background pixel values within cuboids.

4.3.1 Automatic Localization

Weizmann dataset provides the ground truth silhouette action masks for the whole dataset. In reality however, such localization is hard to achieve for realistic datasets. Nonetheless, we designed a system that combines an off the- shelf human detection system [6] with a cuboid saliency method for automatic localization of the action being performed. For the human detection system, we used the system provided online [5] by the authors. We compute masks at a threshold of -4 and pick the best available localization mask. For the saliency detection, we use the system described in [9]. Instead of applying the approach to raw 2-d image data, we use the same method and apply it to descriptors extracted from a video sequence. We thus obtain a region of salient descriptors in the video sequence, construct a bounding box around these descriptor locations and use this as the automatic localization mask. Having computed these two masks, we take their union and use it as an automatic localization of

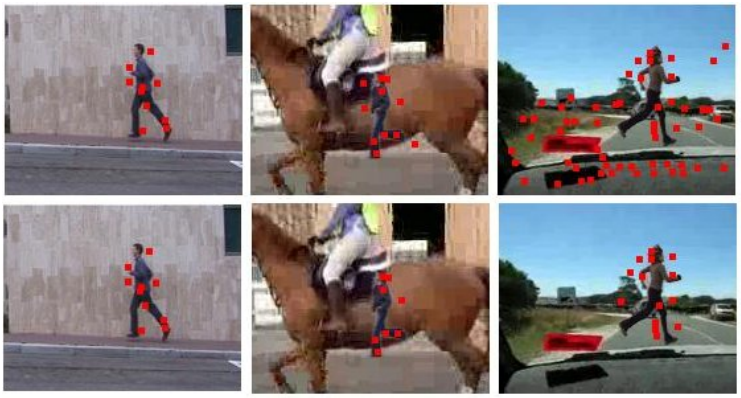

Figure 4.2: Top row shows the intereste points without pruning for Weizman simple, static and Dynamic. Bottom row shows the interest points for the same frame after pruning.

the action. We tested this system on the Weizmann Dynamic dataset as well as the realistic UCF Sports dataset.

4.4 Interest point pruning

Directly running our baseline system on these new synthesized dataset results in interest points detected due to both the action and background motion. Having computed automatic localization information, we can now remove irrelevant background interest points. The goal is to discard all interest points lying outside the automatic localization mask calculated previously. This technique is applied at each frame of the action video sequence. With the removal of these background interest points, the recognition performance is expected to improve.

Figure 4.2 shows the interest points generated for the mentioned dataset. We see that almost all interest points in the original datasets are on or near the person performing the action. For the dynamic dataset however, a significant number of interest points are due to background motion. It is essential that we remove these interest points for improved recognition accuracies. We thus prune interest points lying outside the automatic localization masks generated for this dataset. It should be noted that these localization masks are in fact rectangular bounding boxes and so different from silhouette masks. After pruning, the interest points for the original dataset remain the same. However, interest points from the dynamic dataset are reduced by large extent (see Figure 4.2). Since pruning helps remove irrelevant interest points in the dynamic dataset, we see improvement in recognition results (see Table 4.4). We also present the best possible recognition accuracy that can be achieved using ground-truth localization masks. The ground-truth localization masks are obtained by fitting a tight rectangular bounding box to the action silhouette masks available with the Weizmann dataset.

Although there is improvement in recognition accuracy for the Weizmann dynamic dataset, it is still not comparable to that achieved on the original Weizmann dataset (even when using ground-truth localization). This can be attributed to the presence of background information within the cuboids extracted around the relevant interest points. This background is incorporated in the descriptor construction process and thus negatively affects performance.

Method	Dynamic Weizmann
Our Baseline (Chapter 2)	36.5%
Automatic Localization + Interest Point Pruning	41%
Ground-truth Localization + Interest Point Pruning	68%

Table 4.4: The above table shows the accuracy on synthesized complex dataset when using interest point pruning with automatic localization. Best possible results for interest point pruning with ground-truth localization are also shown. Although results improve, they are still not comparable to those achieved on original Weizmann dataset using our baseline system (Table 2.1).

In the next section, we will discuss actions that are more prune to the presence of background in extracted cuboids and how localization can be used to eliminate this irrelevant information.

4.5 Cuboid Masking

Previously, we showed how generating automatic action localization and using it to prune interest points helps improve recognition accuracy on the new synthesized complex dataset. However, the results obtained are still not comparable to those achieved by baseline systems on original Weizmann dataset (refer to Table 4.5). In this section, we will explore the problem further and show how eliminating background information from within relevant cuboids further improves results.

As stated earlier, moving actions (e.g. running, walking) are more prone to be affected by complex backgrounds than stationary actions (e.g. bending, waving). Despite pruning

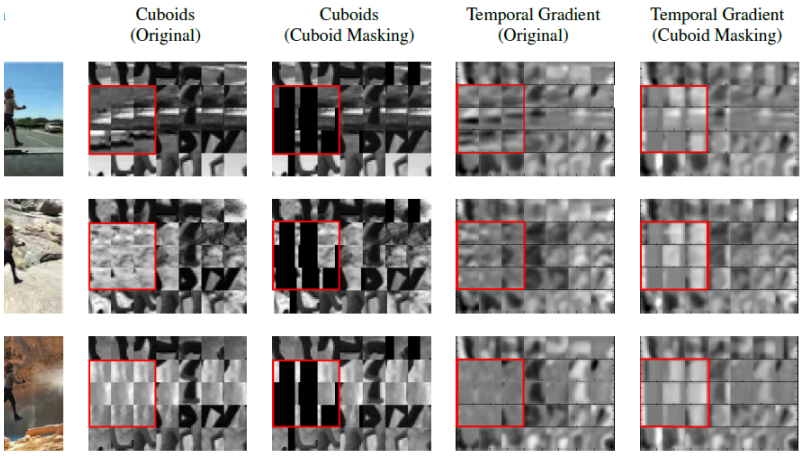

Figure 4.3: The figure shows the effect of cuboid masking. **Column 1:** Shows the same running action performed by the same person matted on 3 different complex moving backgrounds. **Column 2:** Shows cuboids extracted from each video sequence. Size of each cuboid is 13x13x7, where all 7 frames are shown in a single row. **Column 3:** Illustrates the exact same cuboids as in column 2 after applying cuboid masking. **Column 4:** Shows the temporal gradients of cuboids in column 2. **Column 5:** Shows the temporal gradients of cuboids in column 3. The gradient in column 4 corresponding to background content (red outlined) appear different for each video sequence. However, the gradients of all three actions looks similar after applying cuboid masking, as depicted in column 5. This is confirmed by average SSIM values of 0.67 and 0.75 for original temporal gradients (column 4) and cuboid masked temporal gradients (column 5) respectively.

interest points, cuboids may still contain background pixels; cuboids extracted near the mask boundary contain irrelevant spatial information while cuboids extracted for fast moving actions (such as legs of running and walking) contain temporal background information. To deal with this, we need to make use of our automatic localization masks by forcing all pixels of

Method	Dynamic Weizmann
Our Baseline(Section 2)	36.5%
Automatic Localization Interest Point Pruning	41%
Automatic Localization + Interest Point Pruning + Cuboid Masking	48%

Table 4.5: The above table shows the accuracy on Weizmann dynamic dataset using combination of Interest Point Pruning (IPP) and Cuboid Masking (CM) w.r.t **Automatic masks**. We can see that optimal accuracy is achieved when using both IPP and CM strategies.

the extracted cuboids, that lie outside the localization bounding region, to a constant value. This helps *mask* out the irrelevant background pixel values, resulting in similar gradients across same actions in the descriptor construction phase. This modification to the cuboid is what help in optimal results on the new synthesized complex dataset.

An illustration of this is shown in Figure 4.3 for the dynamic dataset. Each row shows the *same* running action performed by the *same* person on *different* dynamic backgrounds. The 2nd column shows some of the extracted cuboids of the corresponding video sequence while the 3rd column shows the same cuboids after applying cuboid masking. The 4th shows temporal gradients corresponding to column 2 while the 5th column shows temporal gradients corresponding to column 3.

For convenience, we highlight cuboid frames showing background pixels in column 2 through 5 with a red outlining. We observe that the background content in the cuboids (column 2) varies significantly for each video, leading to different temporal gradients (column 4)

and eventually different descriptors. Although all 3 videos are of the same action, differences in background force systems to index these videos under different classes and thus decrease overall recognition performance.

On the contrary, application of our cuboid masking technique handles this problem. Column 3 shows how all cuboid frames composed of background content are blackened out. As a result, temporal gradients associated with background information inside cuboids (column 5) are highly similar for each of the action video. This helps in assigning the same label for all 3 videos and thus improve recognition performance.

To strengthen our case, we measure the average structural similarity (SSIM) for temporal gradients with and without cuboid masking of all 3 videos shown in Figure 4.3. We found the average SSIM value to be 0.67 for the case without cuboid masking and 0.75 for the case with cuboid masking. With higher SSIM score, it is evident that cuboids gradients are more similar after cuboid masking and hence improve the recognition results.

Tables 4.5 and 4.5 shows results associated with cuboid masking for both automatic and ground-truth localization. We see an improvement of 11.5% and 52.5% respectively over the baseline results. We can see that even with an average automatic localization method, we are able to achieve more than 10% improvement over the baseline performance. This is a significant jump in performance and shows how cuboid masking is able to handle complex static and dynamic backgrounds. With better localization techniques however, there is scope of even more improvement as depicted by the results obtained using ground-truth localization

Method	Dynamic Weizmann
Our Baseline(Chapter 2)	36.5%
Ground-truth Localization + Interest Point Pruning	68%
Ground-truth Localization + Interest Point Pruning + Cuboid Masking	89%

Table 4.6: The above table shows the accuracy on Weizmann dynamic dataset using combination of Interest Point Pruning (IPP) and Cuboid Masking (CM) w.r.t **Ground truth masks**. We can see that optimal accuracy is achieved when using both IPP and CM strategies.

4.6 UCF Sports

Having analyzed the problem using these synthesized datasets, we next test our system on a realistic dataset. Instead of Youtube [18, 16] and Hollywood [13, 19] datasets, we used the UCF Sports dataset for this task. The reason for this choice being that the UCF Sports dataset is more coherent with regards to the action categories as opposed to both Youtube and Hollywood datasets. In order to show that our solution for recognizing actions on complex backgrounds do not over-fit the synthesized complex datasets, we test our system on the UCF Sports datasets. UCF sports dataset has the complex background and camera movement which were simulated in the synthetic dataset. At the same time, actions are more coherent and well captured unlike Youtube and Hollywood.

The results of different experiments on this dataset are presented in Tables 4.6 and 4.6. We see that automatic localization alone does not improve results but when combined with

Method	UCF Sports
Our Baseline(chapter 2)	69.5%
Automatic Localization +Interest Point Pruning	77%
Automatic Localization +Interest Point Pruning + Cuboid Masking	80%

Table 4.7: The table shows the results on UCF sports with **Automatic mask**. It is evident that IPP and CM strategies improve the accuracy by 12%

Method	UCF Sports
Our Baseline (Chapter 2)	69.5%
Ground-truth Localization + Interest Point Pruning	79%
Ground-truth Localization + Interest Point Pruning + Cuboid Masking	85%

Table 4.8: The table shows the results on UCF sports with **Ground-truth mask**. It is evident that IPP and CM strategies improve the accuracy by 17%

cuboid masking, we see a 12% improvement over the baseline results. We also tested using ground-truth masks for the best possible results and observed a 17% improvement over the baseline results. Using either automatic or ground-truth localization, we observe that application of localization for the purpose of interest point pruning is not sufficient. It is the use of localization to correct cuboid corruption that leads to significant improvement over the baseline method.

CHAPTER 5: DISCUSSION AND CONCLUSIONS

In this paper, we introduce new synthesized, complex datasets which we argue are better suited for analyzing how recognition is affected in presence of background complexity. We show how a change from simple to complex background significantly affects the performance of traditional recognition tools. Using our new synthesized complex datasets, we establish that drop in accuracy is directly related to localization and descriptor formation. A detailed analysis of the new datasets is presented, with special emphasis on the impact of factors such as background gradients, background motion and action localization on the recognition results. In light of the analysis, we show how person localization combined with cuboid modifications helps tackle background complexity problem and thus substantially improve overall recognition results. We show how 'proper' use of localization for interest point pruning and cuboid modification leads to a substantial increase in performance accuracy on both the synthesized and realistic datasets. An automatic localization method is also presented which is shown to outperform the baseline approach. Results are shown with ground-truth masks to show how the good localization helps in improving the recognition accuracy.

LIST OF REFERENCES

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *The Tenth IEEE International Conference on Computer Vision (ICCV'05)*, pages 1395–1402, 2005.

[2] Ivan Laptev and Tony Lindeberg On space-time interest Points In ICCV 2003

[3] M. Bregonzio, S. Gong, and T. Xiang. Recognizing action as clouds of space-time interest points. In *CVPR*, 2009.

[4] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, pages 65–72, 2005.

[5] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/.

[6] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1627–1645, 2010.

[7] P. V. Gehler and S. Nowozin. Let the kernel figure it out: Principled learning of preprocessing for kernel classifiers. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, 06 2009.

[8] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *IEEE 12th International Conference on Computer Vision (ICCV)*, 2009.

[9] S. Goferman, L. Zelnik-Manor, and A. Tal. Context-aware saliency detection. In *CVPR*, pages 2376–2383. IEEE, 2010.

[10] N. Ikizler-Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *ECCV*, 2010.

[11] Z. Jiang, Z. Lin, and L. S. Davis. A tree-based approach to integrated action localization, recognition and segmentation.

[12] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *British Machine Vision Conference*, pages 995–1004, sep 2008.

[13] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[14] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:228–242, 2008.

[15] Z. L. Liangliang Cao and T. S. Huang. Cross-dataset action detection. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 2010.

[16] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:461–468, 2009.

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

[18] J. Liu, Y. Yang, and M. Shah. Learning semantic visual vocabularies using diffusion distance. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:461–468, 2009.

[19] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. *IEEE Conf. Computer Vision and Pattern Recog*, 2009.

[20] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008.

[21] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, page 127, sep 2009.