Automatic Detection of Keyframe and Motion frame in Bharatanatyam Video

by Heeramani Prasad

Submission date: 06-Jun-2020 07:18AM (UTC+0530)

Submission ID: 1337226790

File name: 15CS30015_HeeramaniPrasad_Project_Report_MTP_2.pdf (6.17M)

Word count: 7421

Character count: 36174

Automatic Detection of Keyframe and Motion frame in Bharatanatyam Video

Master Thesis Project report submitted to

Indian Institute of Technology Kharagpur

in partial fulfilment for the award of the degree of

Master of Technology

in

Computer Science and Engineering

by

Heeramani Prasad (15CS30015)

Under the supervision of

Professor Partha Pratim Das

Department of Computer Science and Engineering

Indian Institute of Technology Kharagpur

Spring Semester, 2019-20

May 03, 2020

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

KHARAGPUR - 721302, INDIA

CERTIFICATE

This is to certify that the project report entitled "**Automatic Detection of Keyframe and Motion frame in Bharatanatyam Video**" submitted by **Heeramani Prasad (15CS30015)** to Indian Institute of Technology Kharagpur towards partial fulfilment of requirements for the award of the degree of Master of Technology in Computer Science and Engineering is a record of bonafide work carried out by him under my supervision and guidance during Spring Semester, 2019-20.

Professor Partha Pratim Das
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Kharagpur - 721302, India

Date: May 03, 2020

Place: Kharagpur

Acknowledgements

We want to express our sincere gratitude to our Master Thesis Project guide professor Partha Pratim Das for the constant support for our study and related research, for his persistence, motivation, and extensive experience. His supervision helped in all the time of research and authorship of this report. We could not have believed having a better advisor and guide. Besides our guide, We would like to thank PhD student Mr.Himadri B.G.S. Bhuyan for his insightful comments and assistance to widen our research from various viewpoints.

Thank you.

Contents

List of Figures

List of Tables

Abbreviations

KF	**K**ey **F**rame
MF	**M**otion **F**rame
KP	**K**ey **P**ostures
ICD	**I**ndian **C**lassical **D**ance
HOOF	**H**istogram **O**f **O**ptical **F**low
SVM	**S**upport **V**ector **M**achine
OF	**O**ptical **F**low
LK	**L**ucas **K**anade
TP	**T**rue **P**ositive
TN	**T**rue **N**egative
FP	**F**alse **P**ositive
FN	**F**alse **N**egative
ACC	**A**ccuracy
Min	**Min**imum
Max	**Max**imum
%age	**P**ercentage
hist	**H**istogram
RGB	**R**ed **G**reen **B**lue
fps	**F**rame **P**er **S**econd

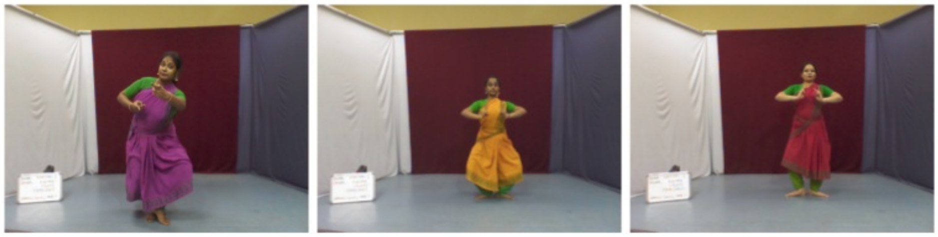
Chapter 1

Introduction

There are following eight forms of dance recognized by the Sangeet Natak Akademi [Wikipedia contributors, 2020d].

1. Bharatanatyam
2. Kathakali
3. Odissi
4. Kathak
5. Kuchipudi
6. Mohiniyattam
7. Manipuri
8. Sattriya

These all dance types represent South Indian religious texts and religious beliefs, especially of Shaivism, Vaishnavism and Shaktism. [Williams, 2004]



(a) (b) (c)

FIGURE 1.1: (a) Dancer 1 (b) Dancer 2 (c) Dancer 3

FIGURE 1.2: Bharatanatyam, Katti kartari Adavu, Variation 1

Bharatanatyam is a important form of IDC (Indian classical dance) that began in the state of Tamil Nadu hundreds of years ago. [Williams, 2004]. Its origin can be traced back to the Natya Sastra. It is a Sanskrit text written by sage Bharata Muni on the performing arts. Bharatnatyam was originated of two words, 'Bharata' and 'Natyam', where 'Bharata' is a mnemonic containing 'bha', 'ra' and 'ta' which respectively. Here 'bhava' is emotion and feelings; 'raga' is a melody, and 'tala' beats. In Hindus temples, Bharatanatyam was exclusive up to the $19^{t}h$ century. In modern days, Bharatanatyam spread out to different parts of India and available on the internet easily.

Bharatanatyam dance movements are portrayed by bent legs, while feet keep beat. In a set of mudras, symbolic hand gestures are used to express a story.

Bharatanatyam have basic choreographic units of a dance sequence. An Adavu is accompanied by percussion and vocal music and follows a specific rhythmic pattern [Mallick et al., 2018].

There are following classification of Adavus in Bharatnatyam. There are multiple variations under Adavus in Bharatnatyam.

1. Tattu Adavu

2. Nattu Adavu

3. Pakka Adavu
4. Kudditu Mettu Adavu
5. Kudditu Nattal Adavu
6. Kudditu Tattal Adavu
7. Paikal Adavu
8. Tei Tei Dhatta
9. Katti Kartari as shown in Figure 1.2
10. Utsanga Adavu
11. Mandi Adavu
12. Sarrikkal Adavu
13. Tirmana Adavu
14. Sarika Adavu
15. Joining Adavu

1.1 Bharatnatyam & Computer Vision

Bharatnatyam contains a large number of motion frames and Key Postures.

In Computer Vision, Human motion, representation and its characteristic is an important topic. They are represented and displayed in various forms through centuries. Human identification using motion analysis is mainly focused on observing human gait. The motion of a person's legs and motion of a person's arms are considered as human gait. There is motion information available in dance frames of

Adavus. So, a high-level feature representation of the dance frame has used using Histogram of optical flow (HOOF).

There are following dancers as shown in Figure 1.2, for Katti Kartari Adavu in Variation 1.

Following important terminology are associated with **Bharatanatyam Dance**:

- **Adavu:** The basic unit of Bharatanatyam.
- **Key Postures(KP):** Momentarily stationary well-defined postures occur within the Adavu
- **Key Frames(KF):** Frames associated with a Key Posture
- **Motion Frames(MF):** Frames associated with motion

1.2 Introduction to Keyframe and motion frame in Bharatanatyam

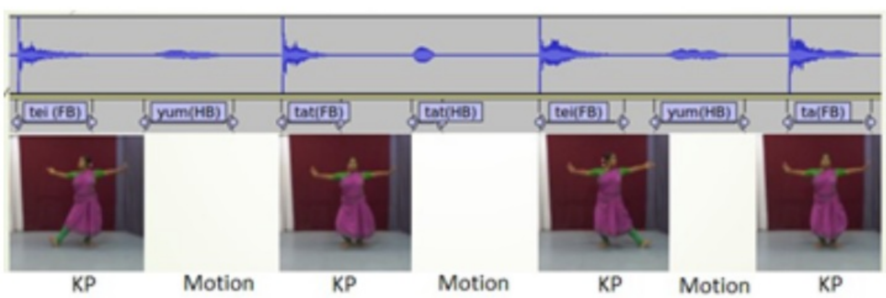


FIGURE 1.3: Occurrence of Motion and KPs

There are various steps involved in Bharatanatyam dance. In Bharatanatyam, the dancer takes momentarily stationary well-defined posture, which occurs within each Adavus followed by sophisticated or complex motions frames. These are called

Key Posture (KP). Then, there are some frames associated with a Key Posture. They are called Key Frames. During the dance, there is a transition from one Key Posture to the next Key Posture. They are called Motion (M). Motions (M) and Key Postures (KP) occur alternately and may repeat in a Bharatanatyam dance performance. Suppose there is Bharatanatyam dance Performance P, which consists of the interleaving sequence of Keyframes and motion frames like K1 M1 K2 M2 K3 M3 ... K(n-1) M(n-1) Kn. Where K is Keyframe, M is a motion frame,1,2,3...,n is $1^st, 2^nd, 3^rd, ..., n^th$ Keyframe or motion frame and n is the total frame. A motion comprised of several set of frames those are not momentarily stationary.

Chapter 2

Literature Survey & Problem Motivation

In this chapter, we will discuss the literature survey, problem motivations and difficulty or challenges faced during project completion.

2.1 Literature Survey

Tanwi Mallick et al. develop NrityaGuru – an autonomous coaching system to give real-time instructional feedback regarding the accuracy of Bharatanatyam as performed by a learner. [Aich et al., 2017]

Manel Sekma et al. proposes motion descriptor called Seg SIFT-ACC for human motion recognising. It is based on temporal segmentation into elementary motion segments. [Sekma et al., 2013]

Matthew Cooper et al. presented approach on temporal video segmentation using supervised machine learning classification. So, we have also tried to use supervised machine learning in the given project. They have created standard features through pairwise similarity of images. [Cooper et al., 2007]

Rizwan Chaudhry et al. suggested a method for representing every frame of a video using a Histogram of oriented optical flow (HOOF). It is used for recognizing human actions by classifying HOOF time-series. [Chaudhry et al., 2009]

Local histogram approaches used local spatiotemporal characteristics or feature for representing human activity in a video. [Laptev, 2005]

Optical flow histograms were applied to match the movement(motion) of a player in a soccer match to that of a subject in a control video. [Efros et al., 2003]

Tran et al. present an optical flow and shape-based approach that uses separate histograms for the horizontal and vertical components of the optical flow as well as the contour of the person as a motion descriptor. [Tran and Sorokin, 2008]

Guozhu Liu at el. showed Key Frame Extraction method from compressed MPEG video data. It helps in reducing Video processing time significantly.It helps in video segmentation and Keyframe extraction. [Liu and Zhao, 2010].

Konečný, J. and Hagara, M used RGB, depth images and combine appearance (Histograms of Oriented Gradients) and motion descriptors (Histogram of optical flow) for parallel temporal segmentation and recognition. [Konečnỳ and Hagara, 2014]

2.2 Motivation

Keyframe, motion frame and Key Posture detection is a fundamental step towards the analysis of dance steps in Bharatanatyam with the perspective of computer vision and human-computer interaction.

- Distinguish Keyframe to motion frame can be used to automate or design an annotation tool. An annotation tool is used to determine which are Keyframe or motion frame
- If the Keyframe is detected, we could recognize Adavu based on the occurrence of key-posture sequences. If the motion frame is detected, we could be able to classify the motion in the given Adavu.

Optical flow is the pattern of the apparent motion of objects, surfaces, and edges in a visual scene created by the relative motion between an observer and a scene [Wikipedia contributors, 2020c]. It is the distribution of ostensible(apparent) velocities of motion of illumination patterns in a scene or image [Wikipedia contributors, 2020c].

- Detection of Keyframe and motion frame from given set frames from Adavus video.
- Optical flow is used to extract the feature from the Gray frame of video as our main objective is to classify the motion frame and Keyframe from frames of video.
- Histogram of optical flow (HOOF) used as a final feature vector to input for binary the classifier.

2.3 Challenges

During the analysis of the keyframe and motion frame, some fundamental difficulties have been faced.

- **Challenges 1**: At the start of the transition. There may be some very slow motions. That slow-motion may be falsely classified as Keyframe.
- **Challenges 2**: The distinction between Keyframe and motion frames may not be accessible due to the existence of complex motions and postures.
- **Challenges 3**: Non-visibility of foot/leg movements and occlusion due to the sophisticated dress style.
- **Challenges 4**: In some cases, when a dancer is in Key posture position, the movement of the dress materials may be misinterpreted as body movements(motion frame).
- **Challenges 5**: The non-availability of annotated Bharatanatyam Adavus.

Chapter 3

Dataset Introduction

In this chapter, we will discuss the dataset and data analysis. Data set for Bharatanatyam Adavus is not open-sourced for research purpose.

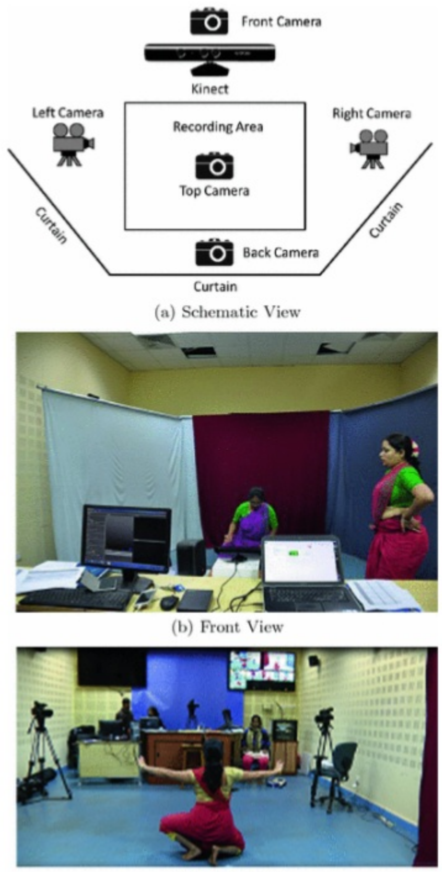
3.1 Dataset Introduction

So, We have recorded different dance formats for Adavus Bharatanatyam with the help of experts, dancers and learners. For the creation of dataset for Bharatanatyam adavus, Microsoft Kinect Wikipedia contributors [2020b] is used to capture RGB, skeleton videos and depth at a rate of 30 frames per second (fps).

The setup used for recording data set is shown in Figure 3.1

In this section of the project, only RGB frames are required for research purpose. Our approach was using a set of standard RGB as input and resized from 480 x 640 to 120 x 160 RGB frames, which had been further converted to grey frames for background subtraction.

The following Adavus had been recorded as a data set.



(a) Schematic View

(b) Front View

(c) Rear View

FIGURE 3.1: Bharatanatyam Studio setup for recording

1. Joining
2. Kartari
3. Nattal
4. Tattal
5. Mandi
6. Mettu
7. Natta

ADAVU	Variant	# Dancers	# Videos	# Frames	# Motion Frame	# Key Frame
Tatta	8	3	24	27155	12531	14624
Natta	8	3	24	18587	9546	9041
Kuditta Mettu	4	3	12	8895	4325	4570
Kuditta Nattal	6	3	18	15743	9816	5927
Kuditta Tattal	5	3	15	20995	13424	7571
Tei Tei Dhatta	3	3	9	5598	4336	1262
Katti Kartari	1	3	3	2449	1747	702
Utsanga	1	3	3	1363	1085	278
Mandi	2	3	6	9975	5449	4526
Tirmana	3	3	9	6415	4181	2234
Sarika	4	3	12	8492	5662	2830
Joining	3	2	6	4460	3110	1350
Total	48		141	130127	75212	54915

TABLE 3.1: Bharatanatyam, Data set introduction

8. Paikal
9. Pakka
10. Sarika
11. Sarikkal
12. Tatta
13. Tei-Tei Dhatta
14. Tirmana
15. Utsanga

Each Adavus have different numbers of variants. For each variation, three different dancers have performed Bharatanatyam dance. The data set has been shown in Table 3.1

After analyzing the dataset, we have observed that the annotation file for some of the Adavus and variation are not available. So, we have removed it from here as it is not relevant for us.

Each Adavu follows a regular pattern; for two consecutive Keyframe, there exist motion frames. All frames before 267, i.e., 1 to 265, are dancer preparing frame

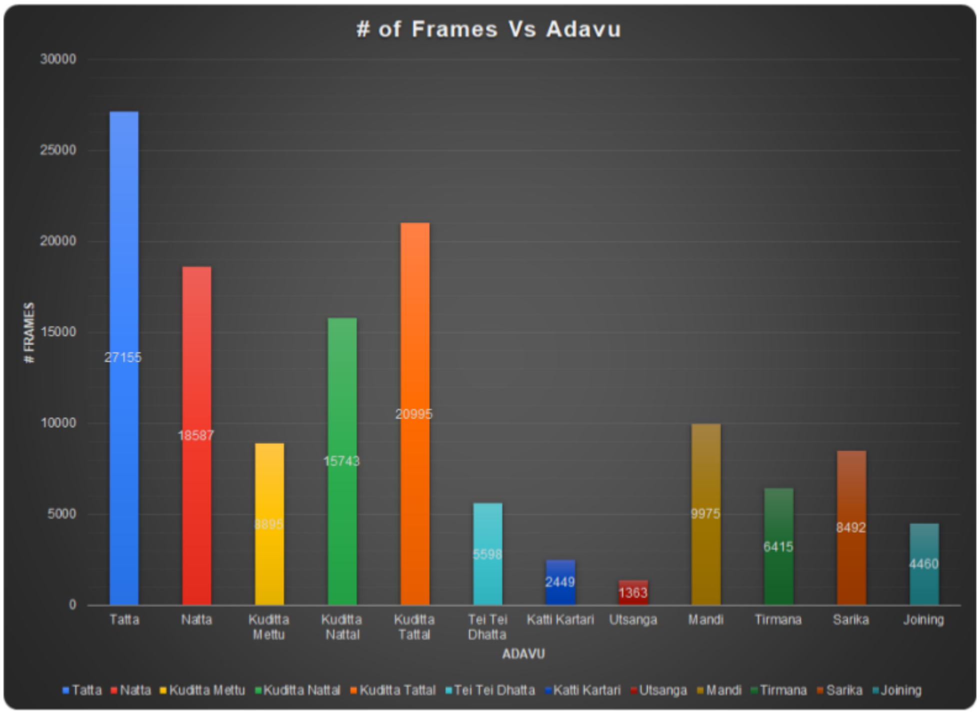


FIGURE 3.2: Bharatanatyam,Pictorial representation Data set

so that we would discard all frames. For example, in the first row, 267 is starting a Keyframe, and 303 is ending the Keyframe. After 303, i.e., from 304 to 314 are motion frames, and again Keyframe would be repeated similarly in Table 3.2

ID	Start Frame #	End Frame #
D1T4P01B1	267	303
D1T4P01B2	315	350
D1T4P01B3	361	396
D1T4P01B4	409	444
D1T4P01B5	458	494
D1T4P01B6	507	542
D1T4P01B7	554	590
D1T4P01B8	603	643
D1T4P01B9	656	684
D1T4P01B10	697	728
D1T4P01B11	742	775
D1T4P01B12	788	821
D1T4P01B13	834	865
D1T4P01B14	879	912
D1T4P01B15	924	953
D1T4P01B16	966	996

TABLE 3.2: Tatta, Variant 4, Dancer 1 annotation file

Chapter 4

Workflow

In this chapter, we will discuss the workflow of our project and some algorithm.

4.1 Workflow

In this section, we will discuss the proposed workflow. It takes RGB frame and outputs labelled feature as Keyfeature**(KF)** and motion frame **(MF)**. Classification of Keyfeature**(KF)** or non-motion frame and motion frame **(MF)** comprises the following parts:

4.1.1 Background Subtraction Module

4.1.1.1 RGB image to Grayscale image:

It takes a colour image(RGB channels) of size 480 x 640. After that, the RGB image is converted to grayscale (single grayscale channel) of size 480 x 640. It helps in decreasing the complexity of the introduced method. Grayscale pixel's value is computed by the weighted sum of the corresponding red, green, and blue pixels as:

$$GrayFrame(i,j) = 0.299 * Frame(i,j)_R$$

$$+0.587 * Frame(i,j)_G$$

$$+0.144 * Frame(i,j)_B$$

$$Where 1 \leqslant i \leqslant 480 and 1 \leqslant j \leqslant 640$$

4.1.1.2 Background Removal

FIGURE 4.1: Redundant information highlighted

Dancer image contains lots of redundant information which is not required for us—for example, background image or board on the side of the dancer, as shown in Figure 4.1.

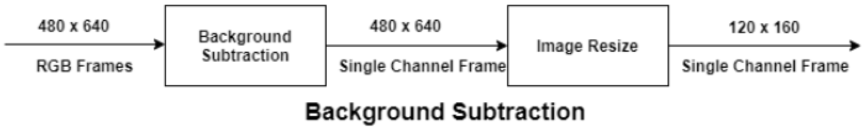


FIGURE 4.2: Background Subtraction

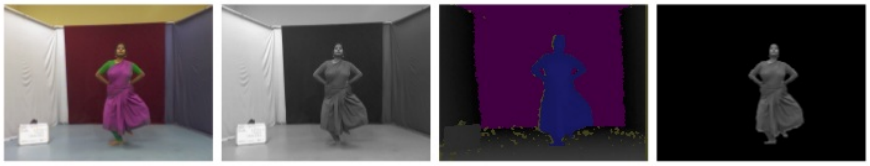
Kinect depth data has 16-bit information, which consists of 3 bits which signify player index and the remaining 13 bits mean depth data. The depth data is the distance between the dancer and the camera lens. It uses the following algorithm as shown in Figure 4.3

Algorithm 1: Background Removal

Input: GrayImage, DepthData
Output: BGSImage // Background subtracted Image
Initialization: Set Mask(i, j) = 0 `// i:1-M, j:1-N`
for i = *1 to M* **do**
  `// Frame size =` $M \times N$
  for j = *1 to N* **do**
    [x, y]=MapDepthToColorFrame(DepthData(i,j))
    if *PlayerIndex(i, j)* > 0 **then**
      Mask(x, y) = 1
BGSImage = GrayImage * Mask
return *BGSImage*

FIGURE 4.3: Background Subtraction algorithm

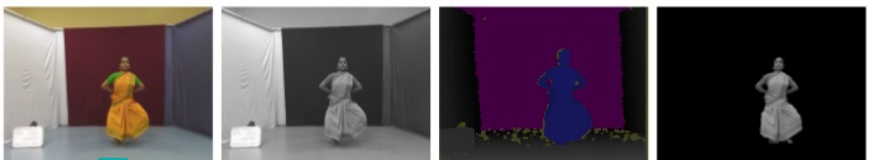
Since we are calculating optical flow which using motion information, so, we would remove the background from a grey image and retain only the information dancer portion. We would use depth stream information from Kinect depth data as shown on Figure 4.2 and 4.3.



(a) (b) (c) (d)

FIGURE 4.4: (a) RGB (b) Grey (c) Depth Frame (d) Single channel

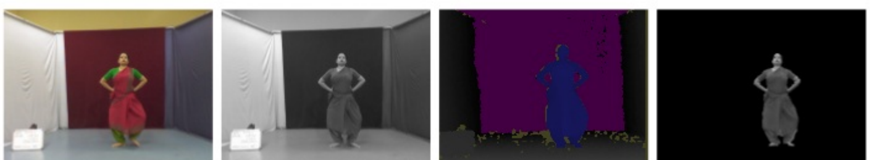
FIGURE 4.5: Tatta->Variation->Dancer 1



(a) (b) (c) (d)

FIGURE 4.6: (a) RGB (b) Grey (c) Depth Frame (d) Single channel

FIGURE 4.7: Tatta->Variation->Dancer 2



(a) (b) (c) (d)

FIGURE 4.8: (a) RGB (b) Grey (c) Depth Frame (d) Single channel

FIGURE 4.9: Tatta->Variation->Dancer 3

4.1.1.3 Resize Image

After that, the image is resized to size 120∗160. We have used OpenCV, `cv2.resize()` function for this purpose. This function takes an image of size $480 * 640$ and resizes the image source down to size $120 * 160$, as shown in Figure 4.2

4.1.2 Preprocessing Example (Backgroud Subtraction)

It is the example of background removal as shown in Figure 4.5, 4.7 and 4.9.

4.1.3 Feature Extraction Module

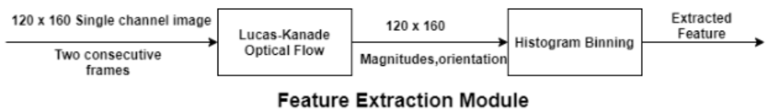


FIGURE 4.10: Feature Extraction

In this section, we will explain the feature extraction method used. This feature would be used for classification KeyFrame and motion frame (non-KeyFrame) in given Bharatnatyam adavu video. Lucas-Kanade optical flow is calculated. Two consecutive frames of size 120 x 160 single-channel images are taken as input. Lucas-Kanade optical flow method yield a matrix of size 120*160, which contains magnitudes and orientations. These are considered as input for Histogram binning. Histogram binning methods depend on the approaches we have taken. These would output the final feature vector for labelling. Different type of Histogram binning is used to feature descriptor.

4.1.3.1 Estimate Optical Flow: Lucas-Kanade Method

'opticalFlowLK' is used to estimate the optical flow using the Lucas-Kanade method [Barron et al., 1994]. This function generates an optical flow object for determining the direction(displacement) and speed of an apparent moving object using the Lucas-Kanade method(optical flow calculation method).

(a) (b) (c)

FIGURE 4.11: (a) Previous Image (b) Next Image (c) Showing Image

4.1.3.2 Algorithms

1. Input: Two consecutive images
2. Output: Optical flow / velocity vector $(V_x, V_y) = (u, v)$ for each pixel
3. Assumption: Brightness/ Intensity doesn't change with time. So, $I(x, y, t) = I(x + u, y + v, t)$

- I_x, I_y, and It are the spatiotemporal image brightness derivatives.
- u is the horizontal velocity vector.
- v is the vertical velocity vector.

$$I_x u + I_y v + I_t = 0$$

4.1.3.3 Lucas-Kanade Method

The Optical flow constraint equation is solved using the Lucas-Kanade for u & v, the method splits the initial image into smaller segments and assumes a consistent velocity in each region. A weighted and least-square fit is performed on the optical flow constraint equation to a consistent model for $[uv]^T$ in each section of Ω. The given process achieves that fit by minimizing following the given equation:

$$\sum_{x \in Q} W^2 [I_x u + I_y v + I_t]^2$$

W is a window function for the center of each section. The solution to the minimization problem is:

$$\begin{bmatrix} \sum W^2 I_x^2 & \sum W^2 I_x I_y \\ \sum W^2 I_y I_x & \sum W^2 I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum W^2 I_x I_t \\ \sum W^2 I_y I_t \end{bmatrix}$$

4.1.3.4 Histogram Binning

Histograms are utilized broadly as non-parametric frequency estimators for visualizing data and for obtaining extract quantities. However, the values of frequency estimators depend on the number of bins taken for the Histogram. There are some rules to determine bins number. For example, Generally, 5-20 bins numbers are satisfactory. As in the case of Matlab, ten(10) bins number is used as a default parameter. We have used following Histogram binning method:

1. Global Histogram using only magnitudes of optical flow.
2. Global Histogram using magnitudes and orientation of optical flow.
3. Fractional Histogram Binning using magnitudes and orientation of optical flow taken over $8 * 8$ cell.

4.1.4 Training Module

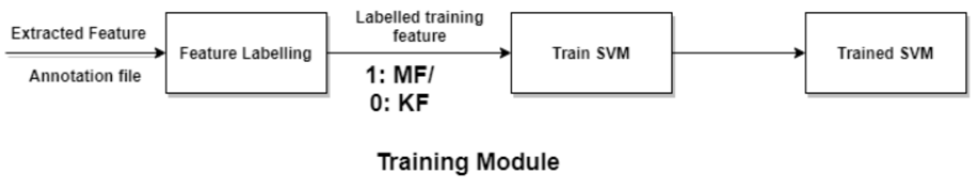


FIGURE 4.12: Training Module

Extracted feature from the previous module with the help of annotation file is used to train the SVM classifier as shown in Figure 4.12.

4.1.4.1 Feature Labeling

Extracted feature and annotation file in 'CSV' format is taken as input for feature labelling. The feature is marked as 1 for motion frame as this is our primary and as 0 for the KeyFrame as shown in Figure 4.12.

4.1.4.2 Training SVM

SVM(Support vector machine) is a mechanism of pattern identification that has a precise theoretical basis. So, we used a linear SVM classifier for feature labelling whether it is KeyFrame or motion frame. Data set means a set of all motion frames and KeyFrame (non-motion frame). The features are generated by Histogram binning in the previous section. The dataset containing all frames was randomly separated into two datasets, i.e., training dataset and test dataset. The training dataset includes 80 per cent, both motion frames, and KeyFrames. Test dataset contains 20 per cent also contain both motion frames and KeyFrames.

```
kernel = 'linear'
#Create a svm Classifier
```

```
clf = svm.SVC(kernel=kernel)  # Linear Kernel

#Train the model using the training sets
clf.fit(X_train, y_train)
```

4.1.5 Testing Module

We would use the test dataset, i.e., the test feature vector is used on trained SVM. This trained SVM would label a given frame, whether it is motion frame(MF:1) or KeyFrame (KF:0) as shown in Figure 4.13.

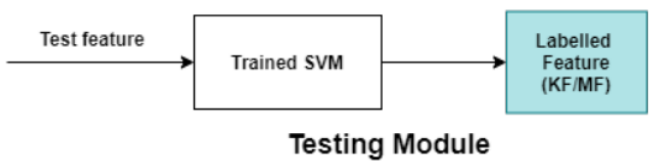


FIGURE 4.13: Testing Module

4.1.5.1 Evaluation metrics

Some motion frames will be detected as KeyFrame and some KeyFrame will be detected as motion frame.

We have used four usually accepted criteria, i.e.,

1. Accuracy
2. Precision
3. Recall
4. F1-Score

It is used for the evaluation of prediction performance of constructed SVM binary classifier models. The accuracy is the number of correctly predicted KeyFrame and motion frame out of the total number of given frames. The precision is the number of actual KeyFrame and motion frame out of predicted KeyFrame and motion frame. The recall is the number of correctly predicted KeyFrame and motion frame out of actual KeyFrame and motion frame The F1 score value is a merged value of recall and precision.

$$F1Score = 2 * \frac{Precision * Recall}{Precisin + Recall}$$

KeyFrame accuracy is determined by following formula:

$$Accuracy_{KF} = \frac{KF_{Detected}}{KF_{Total}}$$

where $Accuracy_{KF}$: Accuracy of KeyFrame,
$KF_{Detected}$: Number of detected KeyFrame &
KF_{Total} : Total Number of KeyFrame.

$$Accuracy_{MF} = \frac{MF_{Detected}}{MF_{Total}}$$

where $Accuracy_{MF}$: Accuracy of Motion Frame,
$MF_{Detected}$: Number of detected Motion Frame &
MF_{Total} : Total Number of Motion Frame.

No doubt we have computed for both motion frame(MF) and KeyFrame(KF). Our objective is to detect motion key(MF).

True Positive(TP) = Motion Frame True Negative(TN) = Key Frame

TP = Motion frame when it was Motion frame.

TN = Key frame when it was Key frame.

FP = Motion frame when it was Key frame.

TN = Key frame when it was Motion frame.

$$Accuracy_{Total} = \frac{KF_{Correct\_Detected} + MF_{Correct\_Detected}}{KF_{Total} + MF_{Total}}$$

Chapter 5

Histogram Binning

In this chapter, we will discuss various histogram used in project.

5.1 Histogram Binning

The Histogram is a primary analytical tool for information analysis, which is accepted as an essential part of various experimental computational algorithms [Shams et al., 2007]. In this section, we analyze multiple approaches used for the creation of a feature vector, as shown in the Figure5.1.

FIGURE 5.1: Testing Module

5.1.1 Approach 1: Global Histogram using only magnitudes of optical flow

In this section, we would discuss a global Histogram using only magnitudes of the optical flow of two consecutive images. We are taking magnitudes part only of optical flow. Since our image size was

$$120 * 160$$

, our input would be of size 19200. We would sort the vector X into bins with intervals defined by the vector $numBins$ (Number of bins). Finally, it would output the frequency of elements in each bin. It would be returned as a feature row vector of size 9. The sample feature vector for two input images is the following:

$$[18611, 173, 142, 115, 71, 50, 29, 4, 5]$$

5.1.2 Approach 2: Global Histogram using magnitudes and Orientation of optical flow

In this section, we would discuss global histogram using magnitudes and Orientation of the optical flow of two consecutive images.

We are taking magnitudes and Orientation of optical flow. Since our image size was $120 * 160$, our input would be two matrices of size $120 * 160$.We would convert $120 * 160$ matrices to 19200 vectors and calculation would be done.

It would create a Histogram of optical flow (HOOF) of the images, based on the respective weightage of magnitude with respect to orientations. Finally, it would output the frequency of elements in each bin. It would be returned as a feature row vector of size 9.

Sample feature vector for two input images is following:

$$[68.98, 5.93, 0, 0, 0, 0, 0, 0, 21.13]$$

5.1.3 Approach 3: Fractional Histogram Binning using magnitudes and Orientation of optical flow taken over 8 x 8 cell

In this section, we would discuss fractional Histogram using magnitudes and orientation of the optical flow of two consecutive images. In this approach, we compute a HOOF descriptor vector for the supplied optical flow of two consecutive images.

We are taking magnitudes and orientation of optical flow. Since our image size was $120 * 160$, our input would be two matrices of size $120 * 160$. It would compute a local histogram of optical flow (HOOF) of $8 * 8$ pixel cells and concatenate to form a 9576 feature vector.

The block size of $2 * 2$ cells is taken. There would be 9 bins for each cell, so $9*4 = 36$ feature vector for a block. There would be a 50% overlap between blocks. So that 15 iterations over the vertical cell and 20 repetitions over the horizontal cell.

$$Feature\_Vector_{\mathbf{Total}} = (15 - 1) * (20 - 1) * 4 * 9 = 9576$$

It would be returned as a feature row vector of size 9.

Sample feature vector of size 9576 for two input images is following:

$$[0.004, 0.002, 0.543, 0, 0, ..., 0.003, 0.006]$$

Chapter 6

Experimental Result

In this chapter, we would discuss various result.

6.1 Global Histogram using only magnitudes of optical flow

In this section, we would discuss the experimental result for Global Histogram using only magnitudes of optical flow. Since we have used only the magnitude part, there is poor accuracy. There is a lot of information loss through orientation flow part.

Adavus	Acc.(MF+KF)	Acc.MF	Acc.KF	precision_mf	precision_kf	f1_mf	f1_kf
Tatta	52.10	66.48	45.78	35.01	75.66	45.86	57.04
Natta	59.26	64.30	55.32	52.92	66.49	58.06	60.39
Kuditta Mettu	71.07	62.24	77.19	65.45	74.66	63.81	75.90
Kuditta Nattal	67.73	62.67	74.47	76.54	60.01	68.91	66.46
Kuditta Tattal	54.72	44.95	69.16	68.30	45.93	54.22	55.20
Tei Tei Dhatta	54.19	45.51	77.69	84.66	34.51	59.20	47.79
Katti Kartari	68.88	63.04	80.00	85.71	53.20	72.65	63.91
Utsanga	43.68	31.11	74.55	75.00	30.60	43.98	43.39
Mandi	72.75	67.04	78.98	77.67	68.72	71.97	73.49
Tirmana	47.90	32.49	73.04	66.28	39.87	43.60	51.59
Sarika	47.81	33.91	70.39	65.06	39.58	44.59	50.67
Joining	65.79	62.11	71.00	75.16	57.01	68.01	63.25
Average	58.82	52.99	70.63	68.98	53.85	57.91	59.09

TABLE 6.1: Used only magnitudes for binning

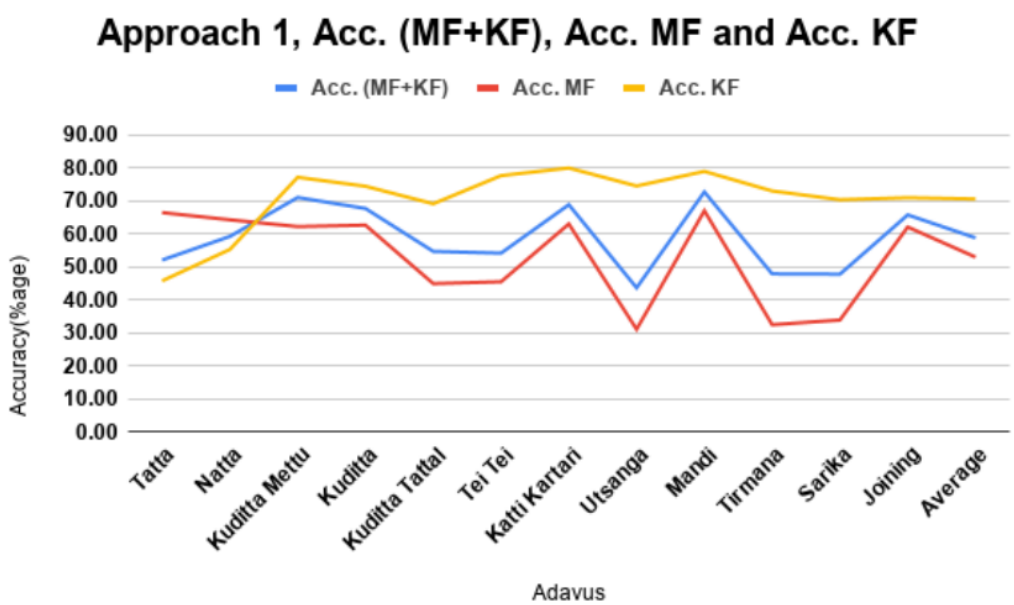


FIGURE 6.1: Approach 1, Acc. (MF+KF), Acc. MF and Acc. KF

1. Accuracy in approach 1 is inheritedly distributed.

2. There may be complex descriptor of KF in Tatta, so there is low accuracy for it.

3. There may cases where Tatta feature could not be represented uniformly in approach 1. As, we are using the only magnitude for histogram binning as shown in the Figure 6.1 and Table 6.1.

6.2 Global Histogram using magnitudes and Orientation of optical flow

In this section, we would discuss the experimental result for global Histogram using magnitudes and orientation of optical flow since we have used the magnitude and orientation of optical flow. There should be better accuracy than better accuracy. We have more information from the orientation of optical flow

Adavus	Acc.(MF+KF)	Acc.MF	Acc.KF	precision_mf	precision_kf	f1_mf	f1_kf
Tatta	71.36	58.02	77.05	51.86	81.16	54.77	79.05
Natta	74.63	67.57	80.42	73.85	75.19	70.57	77.71
Kuditta Mettu	83.3	74.76	89.31	83.1	83.42	78.71	86.26
Kuditta Nattal	77.82	73.5	83.89	86.5	69.27	79.47	75.88
Kuditta Tattal	66.8	57.72	80.4	81.54	55.92	67.59	65.96
Tei Tei Dhatta	64.52	60.57	76.89	89.14	38.36	72.13	51.18
Katti Kartari	71.94	62.65	89.63	92	55.76	74.54	68.75
Utsanga	60.53	56.92	68.33	79.57	42.27	66.37	52.23
Mandi	79.47	76.53	82.84	83.66	75.47	79.94	78.98
Tirmana	62.08	56.39	71.8	77.33	49.11	65.22	58.33
Sarika	53.54	36.79	81.59	76.98	43.55	49.78	56.78
Joining	71.65	68.51	75.61	77.99	65.56	72.94	70.23
Average	69.80	62.49	79.81	79.46	61.25	69.34	68.45

TABLE 6.2: Used magnitudes and Orientation for binning

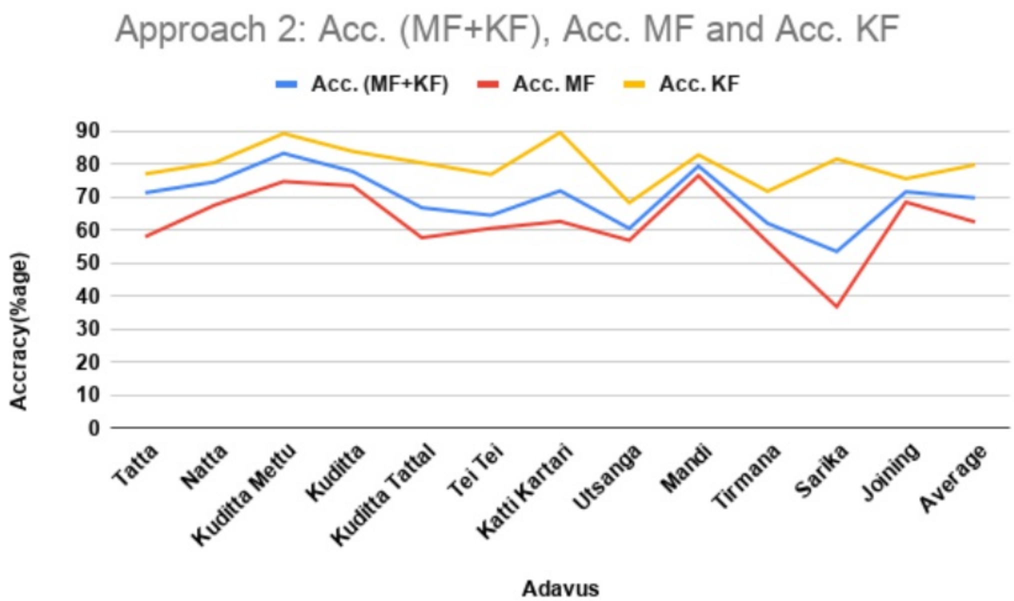


FIGURE 6.2: Approach 2 Acc. (MF+KF), Acc. MF & Acc. KF

6.3 Fractional Histogram Binning using magnitudes and Orientation of optical flow taken over 8 x 8 cell

Adavus	Acc.(MF+KF)	Acc. MF	Acc. KF	precision_mf	precision_kf	f1_mf	f1_kf
Tatta	85.23	80.65	87.25	73.53	91.12	76.92	89.14
Natta	79.05	81.48	77.16	73.59	84.21	77.33	80.53
Kuditta Mettu	79.68	74.88	83	75.36	82.64	75.12	82.82
Kuditta Nattal	76.82	75.54	78.49	82.06	71.12	78.67	74.62
Kuditta Tattal	74.41	81.76	63.55	76.84	70.21	79.22	66.72
Tei Tei Dhatta	78.39	79.82	74.5	89.44	57.72	84.36	65.04
Katti Kartari	80.36	87.16	67.41	83.58	73.39	85.33	70.27
Utsanga	72.63	74.07	69.09	85.47	52.05	79.37	59.38
Mandi	77.78	72.62	83.41	82.68	73.63	77.32	78.22
Tirmana	73.49	73.12	74.1	82.52	62.24	77.54	67.65
Sarika	70.36	69.9	71.1	79.73	59.23	74.49	64.63
Joining	77.97	81.84	72.49	80.78	73.86	81.31	73.17
Average	77.18	77.74	75.13	80.47	70.95	78.92	72.68

TABLE 6.3: Approach 3, Acc. (MF+KF), Acc. MF and Acc. KF

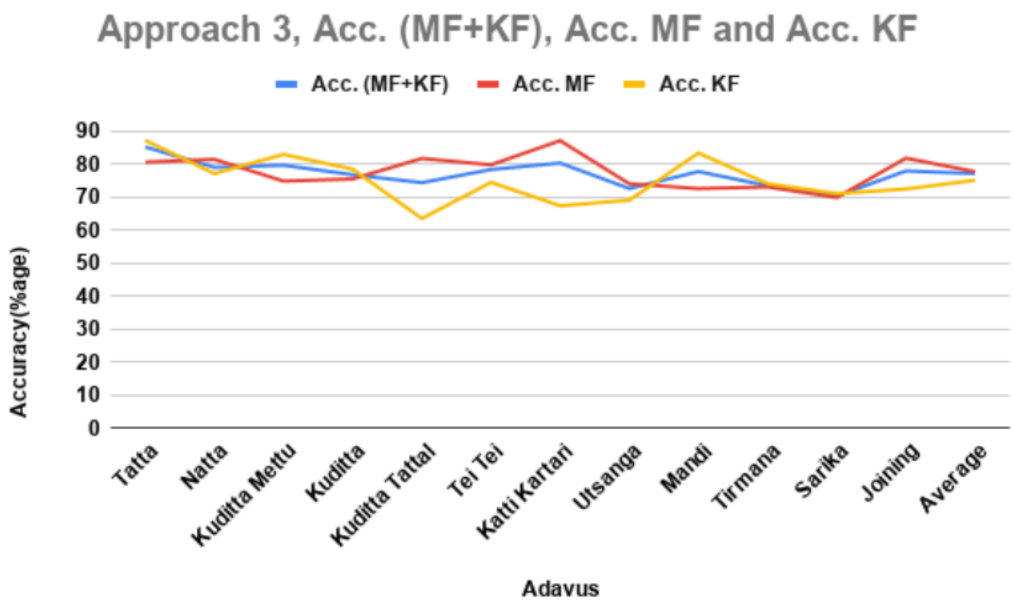


FIGURE 6.3: Approach 3, Acc. (MF+KF), Acc. MF and Acc. KF

1. Approach 3 is better because it does not deviate for $Accuracy_{MF+KF}$, $Accuracy_{MF}$ & $Accuracy_{KF}$ as compare to approach 1 and 2.

2. It could be observed through previous two figures that accuracies non-uniform in approach 1 and 2.

3. In approach 3, It is observed that $Accuracy_{MF+KF}$, $Accuracy_{MF}$ & $Accuracy_{KF}$ for Utsanga does not deviate too much as compared to approach 1.

6.4 Comparison among different Approach 1, 2 & 3

	Accuracy		
Adavus	Approach 1	Approach 2	Approach 3
Tatta	66.48	58.02	80.65
Natta	64.30	67.57	81.48
Kuditta Mettu	62.24	74.76	74.88
Kuditta Nattal	62.67	73.50	75.54
Kuditta Tattal	44.95	57.72	81.76
Tei Tei Dhatta	45.51	60.57	79.82
Katti Kartari	63.04	62.65	87.16
Utsanga	31.11	56.92	74.07
Mandi	67.04	76.53	72.62
Tirmana	32.49	56.39	73.12
Sarika	33.91	36.79	69.90
Joining	62.11	68.51	81.84
Average	52.99	62.49	77.74

TABLE 6.4: MF Accuracy for Approach 1,2 & 3

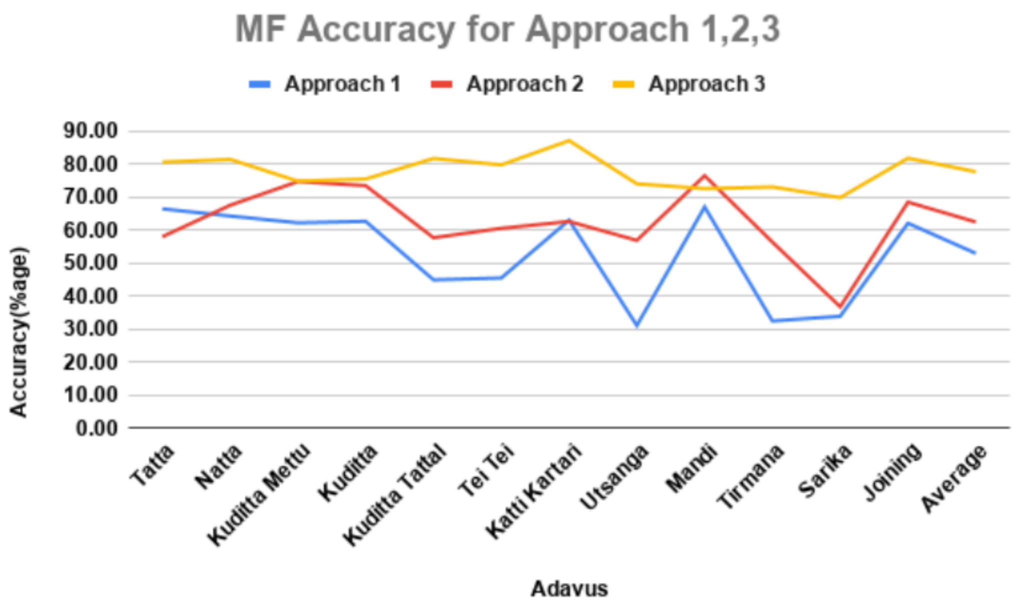


FIGURE 6.4: MF Accuracy for Approach 1,2 & 3

1. For detection of MF from set of adavus frame:

2. Approach 3 is the best approach among the three methods we have tried. Approach 1 is the worst among all methods available.

3. As we know, approach 1 is the simplest. We can see in the figure that $Accuracy_{MF}$ for approach 3 is higher than $Accuracy_{MF}$ for approach 1 and approach 2.

4. Approach 3 has fine detail about optical flow, and it is normalized for every cell so it perform better than others.

5. We could also observe that approach 1 and 2 are performing inadequate but approach 3 performance was good, specially in case **teitei dhatta** adavu, **katti kartari** adavu, **utsanga** adavu and **sarika** adavu. It could be seen that sarika adavu accuracy gap is wide.

6. Overall approach 3 perform better than method 1 and 2.

	Accuracy		
Adavus	**Approach 1**	**Approach 2**	**Approach 3**
Tatta	45.78	77.05	87.25
Natta	55.32	80.42	77.16
Kuditta Mettu	77.19	89.31	83.00
Kuditta Nattal	74.47	83.89	78.49
Kuditta Tattal	69.16	80.40	63.55
Tei Tei Dhatta	77.69	76.89	74.50
Katti Kartari	80.00	89.63	67.41
Utsanga	74.55	68.33	69.09
Mandi	78.98	82.84	83.41
Tirmana	73.04	71.80	74.10
Sarika	70.39	81.59	71.10
Joining	71.00	75.61	72.49
Average	70.63	79.81	75.13

TABLE 6.5: KF Accuracy for Approach 1,2 & 3

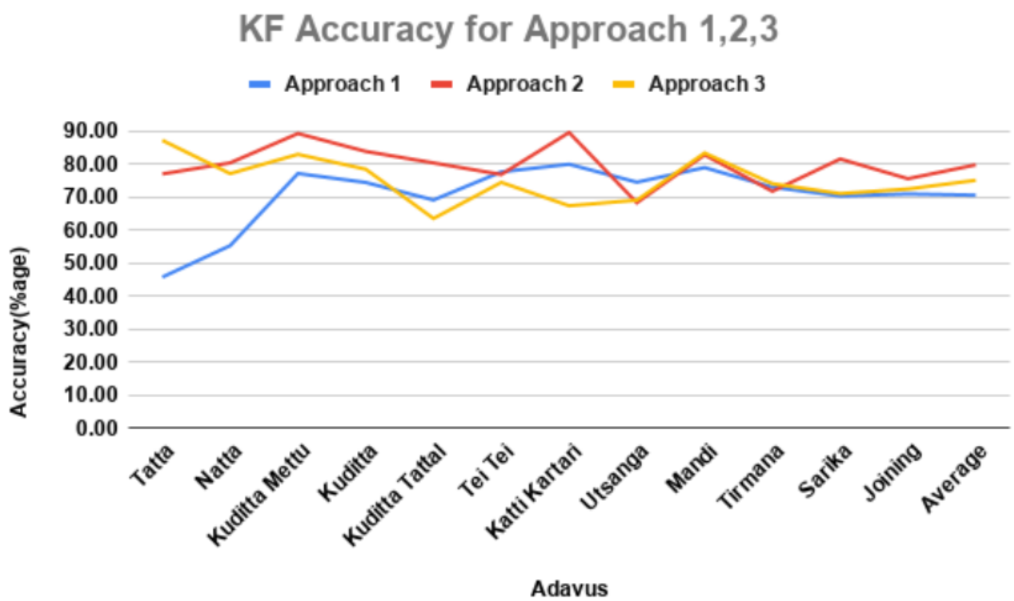


FIGURE 6.5: KF Accuracy for Approach 1,2 & 3

1. Analysing in the figure KF Accuracy for approach 1, 2 & 3. We can say that when Approach 2 & 3 is used, then accuracy improved for **Tatta** as compared to approach 1.(see in figure Approach 1, Acc.MF+KF, Acc.MF and Acc.KF). 6.1

	Accuracy		
Adavus	**Approach 1**	**Approach 2**	**Approach 3**
Tatta	52.10	71.36	85.23
Natta	59.26	74.63	79.05
Kuditta Mettu	71.07	83.30	79.68
Kuditta Nattal	67.73	77.82	76.82
Kuditta Tattal	54.72	66.80	74.41
Tei Tei Dhatta	54.19	64.52	78.39
Katti Kartari	68.88	71.94	80.36
Utsanga	43.68	60.53	72.63
Mandi	72.75	79.47	77.78
Tirmana	47.90	62.08	73.49
Sarika	47.81	53.54	70.36
Joining	65.79	71.65	77.97
Average	58.82	69.80	77.18

TABLE 6.6: MF+KF Accuracy for Approach 1,2 & 3

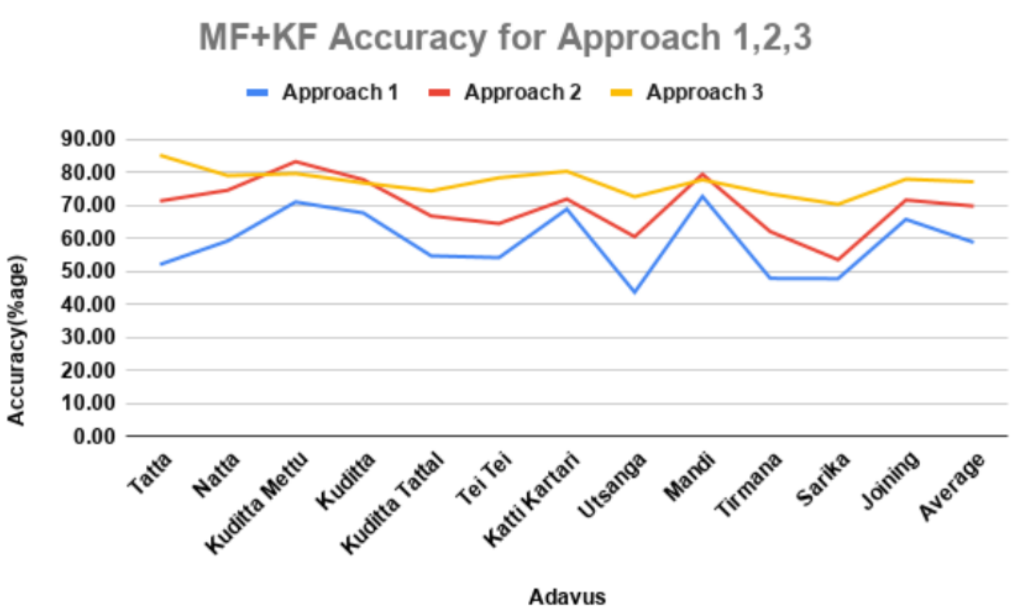


FIGURE 6.6: MF+KF Accuracy for Approach 1,2 & 3

1. $Accuracy_{MF+KF}$ is the weighted average of $Accuracy_{MF}$ & $Accuracy_{KF}$. So, obivoulsy it depends on the $number_{M}F$ and $number_{K}F$ presents in each adavus.

Adavus	Acc. KF Curr.	Acc. KF Prev
Tatta	87.25	98.813
Natta	77.16	94.485
Kuditta Mettu	83	95.67
Kuditta Nattal	78.49	95.054
Kuditta Tattal	63.55	92.092
Tei Tei Dhatta	74.5	89.971
Katti Kartari	67.41	95.118
Utsanga	69.09	81.871
Mandi	83.41	90.614
Tirmana	74.1	73.718
Sarika	71.1	86.442
Joining	72.49	92.846
Average	75.13	90.56

Table 6.7: Approach 3 Acc. KF Curr. VS Acc. KF Prev

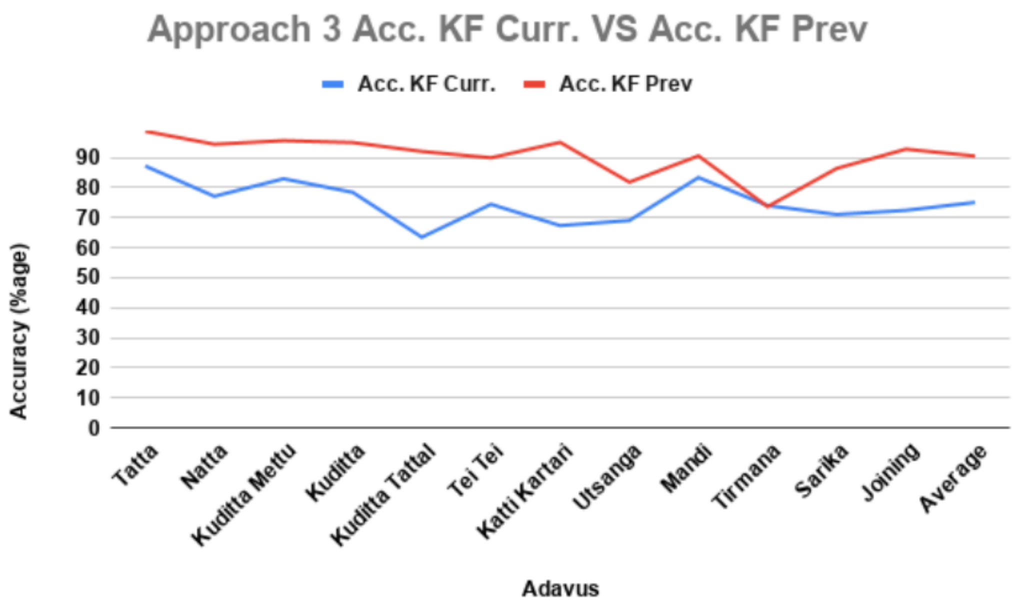


FIGURE 6.7: Approach 3 Acc. KF Curr. VS Acc. KF Prev

1. No doubt we have computed accuracy for both MF and KF. But our main objective is to detect MF and hence tried to optimize MF accuracy. Here, I have compared KF accuracy with "Segmentation of Bharatanatyam dance video: detection of keyframe" as done in the paper by Himadri.

6.5 Minimum accuracy plot

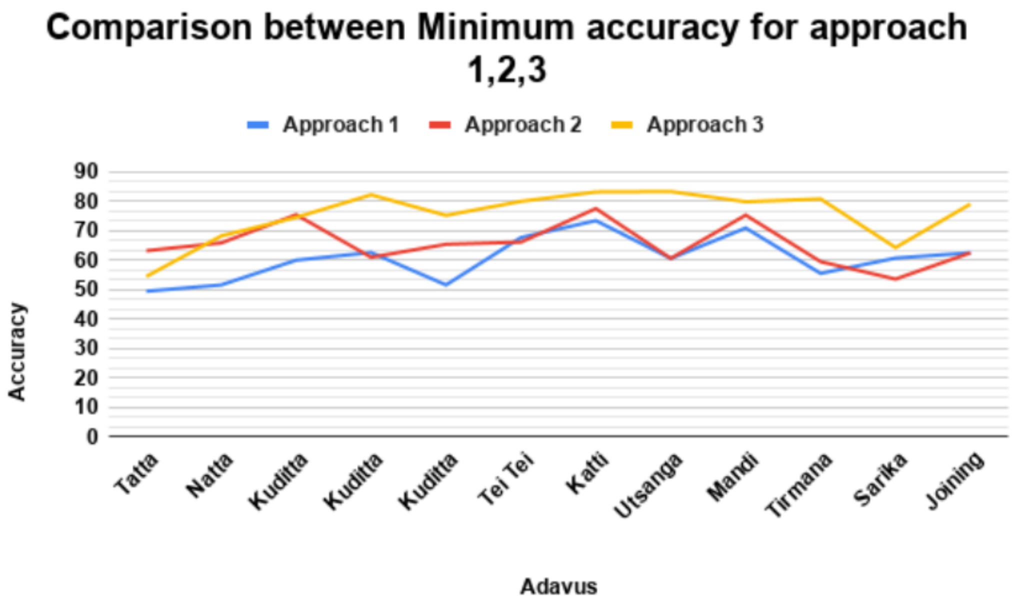


FIGURE 6.8: Minimum accuracy plot

6.6 Maximum accuracy plot

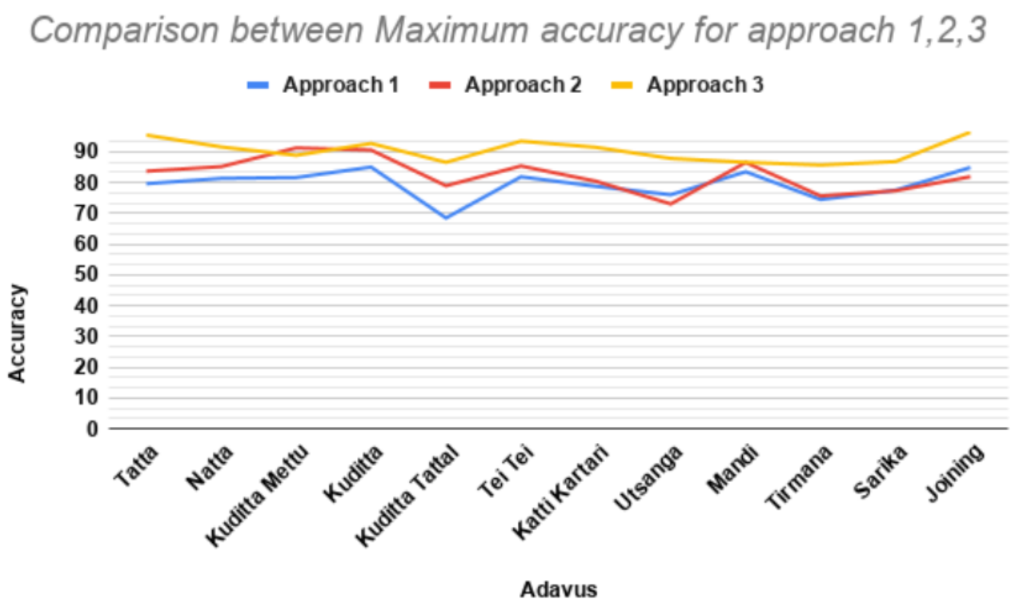


Figure 6.9: Maximum accuracy plot

Chapter 7

Conclusion

7.1 Conclusion

The proposed approach could successfully detect Keyframe and motion frame from given set frames from Adavus video. Optical flow help in the extraction of the feature.

Histogram of optical flow (HOOF) and histogram binning successfully helps in feature vector extraction. It is used as a final feature vector to train the binary SVM classifier. Approach 3 gives the best accuracy among all approaches because we are covering the whole image by block by block. It helps in the preservation of information and optical flow of the image.

Chapter 8

Acknowledgement

8.1 Acknowledgement

This project authors fully acknowledge the support from Prof. Partha Pratim Das, Head, Rajendra Mishra School of Engineering Entrepreneurship, and Ph.D. student Himadri B.G.S. Bhuyan. We have taken immense help from MatLab software and its support community.

Bibliography

1. Achyuta Aich, Tanwi Mallick, Himadri BGS Bhuyan, Partha Pratim Das, and Arun Kumar Majumdar. Nrityaguru: A dance tutoring system for bharatanatyam using kinect. In *National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics*, pages 481–493. Springer, 2017.

2. John L Barron, David J Fleet, and Steven S Beauchemin. Performance of optical flow techniques. *International journal of computer vision*, 12(1):43–77, 1994.

3. Rizwan Chaudhry, Avinash Ravichandran, Gregory Hager, and René Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1932–1939. IEEE, 2009.

4. Matthew Cooper, Ting Liu, and Eleanor Rieffel. Video segmentation via temporal pattern classification. *IEEE transactions on multimedia*, 9(3):610–618, 2007.

5. Alexei A Efros, Alexander C Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *null*, page 726. IEEE, 2003.

6. Jakub Konečnỳ and Michal Hagara. One-shot-learning gesture recognition using hog-hof features. *The Journal of Machine Learning Research*, 15(1):2513–2532, 2014.

7. Ivan Laptev. On space-time interest points. *International journal of computer vision*, 64(2-3):107–123, 2005.

8. Guozhu Liu and Junming Zhao. Key frame extraction from mpeg video stream. In *2010 Third International Symposium on Information Processing*, pages 423–427. IEEE, 2010.

9. Tanwi Mallick, Partha Pratim Das, and Arun Kumar Majumdar. Characterization, detection, and synchronization of audio-video events in bharatanatyam adavus. In *Heritage Preservation*, pages 241–268. Springer, 2018.

10. Anindhya Sankhla, Vinanti Kalangutkar, Himadri BGS Bhuyan, Tanwi Mallick, Vivek Nautiyal, Partha Pratim Das, and Arun Kumar Majumdar. Automated translation of human postures from kinect data to labanotation. In *National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics*, pages 494–505. Springer, 2017.

11. Richard Schechner. *Between theater and anthropology*. University of Pennsylvania press, 2010.

12. Manel Sekma, Mahmoud Mejdoub, and Chokri Ben Amar. Human action recognition using temporal segmentation and accordion representation. In *International Conference on Computer Analysis of Images and Patterns*, pages 563–570. Springer, 2013.

13. Ramtin Shams, RA Kennedy, et al. Efficient histogram algorithms for nvidia cuda compatible devices. In *Proc. Int. Conf. on Signal Processing and Communications Systems (ICSPCS)*, pages 418–422. Citeseer, 2007.

14. Apratim Sharma. Recognising bharatanatyam dance sequences using rgb-d data. *IIT, Kanpur, India*, 2013.

15. Du Tran and Alexander Sorokin. Human activity recognition with metric learning. In *European conference on computer vision*, pages 548–561. Springer, 2008.

16. Wikipedia contributors. Bharatanatyam — Wikipedia, the free encyclopedia, 2020a. URL `https://en.wikipedia.org/w/index.php?title=Bharatanatyam&oldid=956699282`. [Online; accessed 31-May-2020].

17. Wikipedia contributors. Kinect — Wikipedia, the free encyclopedia, 2020b. URL https://en.wikipedia.org/w/index.php?title=Kinect&oldid=958970348. [Online; accessed 1-June-2020].

18. Wikipedia contributors. Optical flow — Wikipedia, the free encyclopedia, 2020c. URL https://en.wikipedia.org/w/index.php?title=Optical_flow&oldid=956430495. [Online; accessed 2-June-2020].

19. Wikipedia contributors. Sangeet natak akademi — Wikipedia, the free encyclopedia, 2020d. URL https://en.wikipedia.org/w/index.php?title=Sangeet_Natak_Akademi&oldid=952792162. [Online; accessed 4-June-2020].

20. Drid Williams. In the shadow of hollywood orientalism: Authentic east indian dancing. *Visual Anthropology*, 17(1):69–98, 2004.

Automatic Detection of Keyframe and Motion frame in Bharatanatyam Video

ORIGINALITY REPORT

13%	9%	5%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Indian Institute of Technology, Kharagpure Student Paper	2%
2	www.cis.jhu.edu Internet Source	1%
3	Submitted to University of Edinburgh Student Paper	1%
4	la.mathworks.com Internet Source	1%
5	de.mathworks.com Internet Source	1%
6	link.springer.com Internet Source	1%
7	Submitted to Institute of Technology, Nirma University Student Paper	1%
8	dr.ntu.edu.sg Internet Source	<1%

9	"Computer Vision, Pattern Recognition, Image Processing, and Graphics", Springer Science and Business Media LLC, 2018 Publication	<1%
10	en.wikipedia.org Internet Source	<1%
11	www.datacamp.com Internet Source	<1%
12	www.oalib.com Internet Source	<1%
13	Submitted to University of Arizona Student Paper	<1%
14	Submitted to University College London Student Paper	<1%
15	www.mdpi.com Internet Source	<1%
16	www.canberra.edu.au Internet Source	<1%
17	Submitted to Higher Education Commission Pakistan Student Paper	<1%
18	Submitted to University of Stellenbosch, South Africa Student Paper	<1%

19	Li Li. "Top-Down Cues for Event Recognition", Lecture Notes in Computer Science, 2011 Publication	<1%
20	"Computer Analysis of Images and Patterns", Springer Science and Business Media LLC, 2013 Publication	<1%
21	Paisitkriangkrai, S., C. Shen, and J. Zhang. "Performance evaluation of local features in human classification and detection", IET Computer Vision, 2008. Publication	<1%
22	Submitted to iGroup Student Paper	<1%
23	users.cecs.anu.edu.au Internet Source	<1%
24	Submitted to Queensland University of Technology Student Paper	<1%
25	Submitted to Rajarambapu Institute of Technology Student Paper	<1%
26	Jaydeb Mondal, Malay Kumar Kundu, Sudeb Das, Manish Chowdhury. "Video shot boundary detection using multiscale geometric analysis of nsct and least squares support vector machine",	<1%

Multimedia Tools and Applications, 2017
Publication

27	"Communications, Signal Processing, and Systems", Springer Science and Business Media LLC, 2020 Publication	<1%
28	eprints.mdx.ac.uk Internet Source	<1%
29	uu.diva-portal.org Internet Source	<1%
30	Rui Sarmento, Vera Costa. "chapter 3 Dataset", IGI Global, 2017 Publication	<1%
31	Submitted to Royal Melbourne Institute of Technology Student Paper	<1%
32	eudl.eu Internet Source	<1%
33	agpu.net Internet Source	<1%
34	Submitted to University of Warwick Student Paper	<1%
35	Submitted to Napier University Student Paper	<1%

Submitted to University of Melbourne

36	Student Paper	<1%
37	www.ece.nus.edu.sg Internet Source	<1%
38	cas.ee.ic.ac.uk Internet Source	<1%
39	www.ijarcce.com Internet Source	<1%
40	chiryo.phar.nagoya-cu.ac.jp Internet Source	<1%
41	Submitted to University of Pittsburgh Student Paper	<1%
42	Jui-Yu Wu. "Self-Organizing Polynomial Neural Network for Forecasting Chaotic Time Series: Comparison Results", Communications in Computer and Information Science, 2011 Publication	<1%
43	Submitted to Amrita Vishwa Vidyapeetham Student Paper	<1%
44	"Heritage Preservation", Springer Science and Business Media LLC, 2018 Publication	<1%
45	Submitted to University of East London Student Paper	<1%