Classical Machine Learning applied to surgical knot analysis

Khushboo Goyal B.Tech

A Dissertation

Presented to the University of Dublin, Trinity College

in partial fulfilment of the requirements for the degree of

Master of Science in Computer Science (Future Networked Systems)

Supervisor: Dr.Gerard Lacey

August 2019

Declaration

I, the undersigned, declare that this work has not previously been submitted as an exercise for a degree at this, or any other University, and that unless otherwise stated, is my own work.

Khushboo Goyal

August 13, 2019

Permission to Lend and/or Copy

I, the undersigned, agree that Trinity College Library may lend or copy this thesis upon request.

Khushboo Goyal

August 13, 2019

Acknowledgments

Throughout the dissertation, I have gotten a lot of excitement, support and help. I want to thank my supervisor, Dr Gerard Lacey of the SCCS at Trinity College Dublin, Ireland, for the continuous support of my Master's dissertation and research, for his understanding, inspiration and monstrous learning. Throughout the dissertation he was so kind and humble to answered all my questions related to thesis or any career advice.I would also like to thank my mentor, Jhilik Bhattacharya from Thapar University (India), for her valuable guidance at every point of time from collecting data till helping in implementation. Without her guidance, I was not able to complete any of the parts of my thesis.I don't think without my parents support I will do anything in my life, they have given me precious advice all the time. They are not living with me but was there for me every point of time. I want to thank my parents, whose adoration and direction are with me in whatever I pursue. They are a true role model for me. Above all, I wish to thank my closest companions Ankur Aghi, Yachana Bhatiya and Vishakha Singhal, who provide unending inspiration.

Khushboo Goyal

University of Dublin, Trinity College
August 2019

Classical Machine Learning applied to surgical knot analysis

Khushboo Goyal, Master of Science in Computer Science

University of Dublin, Trinity College, 2019

Supervisor: Dr.Gerard Lacey

This thesis focuses on the automated framework for video-based surgical knot assessment that is building on the work of the Irfan Essa's paper "Video Based Assessment of OSATS Using Sequential Motion Textures". He showed assessments using the OSATS (Objective Structured Assessment of Technical Skills) criteria, whereas this thesis focuses on the following aspects of surgical motion. Our surgical knot tying hand gesture system divides the video into two classes forefinger throw and backhand throw. The framework is tested on different levels performing basic surgical knot on knot tying kit. This thesis demonstrates the extraction of features like HOG-HOF from the video, compressed by k-means cluster and pass it to SVM with different kernel

Subsequent to extricating the features or keypoints for each video after using the Saptio-temporal Interest Point (STIPs), a vector quantization method maps key points or features from each video into a unified dimensional histogram vector (bag-of-words) after K-means clustering. This histogram is treated as an feature vector for a binary class SVM to fabricate the classifier where data is divided into 80:20(training, testing) dataset and also performs 3-fold cross-validation on the dataset.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this part, the topic of the dissertation is presented, action recognition in videos. Presented inspiration of work, the scientific categorization, which is utilized in this work, and problem statement. Then examined research challenges, identified with action recognition and in the end primary contribution in this point. At last closes with thesis structure.

1.1 Introduction

This dissertation aims to explore gesture recognition to track hands to identify the surgical knot accuracy. The system was implemented based on Computer Vision approaches and Machine learning algorithms, with the help of RGBD camera and high GPU processor.Our mind works on the information based on visuals and process it. A picture contains thousand of words or images. Thus, one frame of the video contains generally 1.8 million of words which credited to Dr James McQuivery of Forrest Research. We, for the most part, comprehend a scene by means of video which gives general data, and from many years people are transmitting information via videos for better understanding.

As technology increases, the number of devices introduced in the market with better quality video feature embedded in low prices. With an increase in the growing of the video, the need for understanding also increases. The human capability of the understanding video in a natural way is less than an intelligent system, that could

analyze it or recognize the actions in the video. Gestures are the powerful means of communication in human life. If we talk about army, soldiers communicate with each other with the help of hand gestures and due to poor light or obligations it is difficult in communication, so Computer Vision helps in recognizing gesture activity and send to fellow soldiers. This is one example which uses Computer Vision, there are many fields in which Computer Vision are used.[10]

There are three main kind of methods which is used to recognize gesture, first method is based on sensor gloves which recognize gesture very well but requires user to wear lead cables and additional devices. Another method is based on video-based recognition which requires camera and hands for gesture, in this case users are not required to wear devices and easy to perform without any obligations.Third and the last method is 3D model-based hand method, this technique recognizes hand correctly but requires more time in compare to other methods.[11]

Some of the application domain for hand gesture which are as follows-

1. Virtual Reality (VR) Applications- Now a days gesture recognition takes one of the greatest achievement in computer science. In these applications, gestures are used to give realistic behaviour of virtual objects for 2D and 3D displays.[11]

2. Laptops and PCs/Tablets-Motions can give decent communication to mouse and keyboard. Many applications consist of manipulation graphics, editing and commenting using pen based gestures.[11]

3.Games- If we talk about in gaming sections there are many researchers who invented gesture based video games based on body movement or hand gestures like driving cars etc for ease and entertainment. Also, PS2 invented one game called Eye Toy, a camera which tracks hand movement for interactive games.[11]

4. Sign Language- As we discussed already gesture movements(very structural and suitable for benchmark for vision algorithms are useful for the people who are disable and understands only sign language like deaf, CV helps to make them understand in a better way. [11]

5. Robotics- As in the growth of technology, people are more curious to research on robots and to interact with robots hand or body gesture is very necessary. It helps robots to reach and manipulate actual real life entity with its movement via a world.[11]

1.2 Motivation

There are many powerful machines which are already in the market to recognize gesture or human action. The crucial primary role of this application is in the Medical field where surgeons are busy in surgery and are not able to help novice surgeon for the knot tying process. High-quality surgical skills development is time-consuming for the surgeon to teach a perfect knot, requiring expert surgeon supervision and evaluation for the whole process in all stages. The manual assessments are difficult for both hospitals and colleges in terms of resources for training.[12]

Figure 1.1: Surgical knot tying
[13]

Many exams systems are used in the medical colleges or hospital to reduce the manual work which covers several criteria like time and motion, knowledge of the procedure, the flow of operation, instruments handling, knot tying, overall performance, etc. [14] Surgery is a very tough task which includes knot tying and suturing of the tissue that involves repetitive hand movements. Before performing major surgery, every master surgeon practices the knot tying process or necessary skills many times for better results and perfection. Now if we consider many surgeons perform the same process and go through OSATS(Objective Structured Assessment of Technical Skills) exam test, then it requires more resources and energy and automated assessments which benefits medical schools and hospitals.

1.3 Taxonomy and Problem Statement

The main idea of this dissertation is to track hands while knot tying process. This thesis can be performed at various degrees of reflection. The most recent couple of years, numerous scientific taxonomies have been proposed to recognize the action performed by people. The process of knot tying- one hand is categorized into two motions or gestures:

Fore figure throw- In this type of knot fore figure is used to perform a knot with one hand.

Back hand throw- In this knot, one hand will be in the first direction, and only the middle figure performs a knotting process.

Others- All the other knots which are not fft and bht are considered as error or other knot.

1.4 Thesis Structure

Chapter 2 provides an overview of the field and the related work which is relevant to my work. In Chapter 3, core background concepts are discussed. Chapter 4 provides the design of the system, while Chapter 5 covers its implementation. Finally, Chapter 6 discuss the future work which expands upon my findings.

Chapter 2

Related Work

The history of Computer Vision shows the advancement of technology in a shorter time. With the help of Computer Vision, we can process any image, photogrammetry, and graphics are possible.

It seemed an unrealistic task for a scientist at the beginning of the 20th century. They were not fully equipped, so processing or editing of images was impossible, and all the image processing activity needed manual work. After a group of people led by Allen Newell investigated the connection between an electronic device and data analysis and came up with the first AI program which helped computers to play checker, speak English language and some simple Maths questions. After that, Larry Roberts extracted 3D information from a 2D image by "block centred" approach. The first success in the field of computer vision came in year 1950s to 1960s.[15]

AI Winter was the term used to describe the time when AI was criticized and underfunded. Lack of scientist and equipment's support from International market individuals work in groups. Even after all these problems, Kurzweil invented Optical character Recognition program, which helps the handicap of blindness. [15]

In the 1980s, Marr was trying to apply edge detection and image segmentation techniques to the original image to create a 2D sketch and 2.5D model. Later he came up with 3D image model, but his techniques were very hard and not easy to implement which was the biggest problem at that time, but his work considered as the significant breakdown in the field of Computer Vision. In the 1990s, Mattew Turk and Alex created an automatic face recognition system based on probability and stats which was

used to identify a face and also generated new faces. However, human interference is needed to train computers even if computer vision history is demonstrated for image tagging and video tracking.[15]

2.1 Hand Pose Recognition

Hand gesture consist of many expression expressed by body includes face expression and hands movement. Among all the gestures and expression, hands poses are the most useful ones in terms of disabled people and also to enable more easy communication with virtual reality system hand poses proved excellent results. [16] There are mainly two hand gesture recognition techniques, first is based on vision-based, and the other one is based on hand gloves data. This hand glove is made from the sensors which convert physical movements of the hand to electrical signals which determine the pose of hand or identifies the subject's hand via colour segmentation.

Hand segmentation is principally worried about skin detection and standardization, usually handled with parametric(Gaussian demonstrating) and non-parametric (histograms) methods. However, its simplicity comes at the cost of little robustness, as non-dynamically adapting models are sensitive to lighting conditions and skin tones[17]. To enhance maximum performance, feature extraction is used in cases like the contour of the hand, location of fingers and palm coordinates etc. In the paper [18], a feature vector is used for brightness pixels and the ratio of the bounding box that contains hand. Center of gravity of segmented hand, distance from the location was taken to construct a circle and extract a binary signal to check the number of active fingers. [19]

2.2 Surgical Skill Recognition

There are two domains which determine the surgical skills, first is with the help of Robotic Minimally-invasive surgery(RMIS) for example, da-Vinci and second is manually where assessment tests are conducted to determine the skills. With the help of RMIS, all the tasks are performed by the robots, which identifies the recognition of surgical gestures using robotic data.

Datta et al. [20] predicted the efficiency score as the ratio of number id detected hand movements and OSATS. The results signify the relationship between surgical efficiency and OSATS rating. The correlation between hand movements and individual OSATS score was not predicted.

Figure 2.1: da vinici [1]

2.3 Local Feature Methods

This technique is also using nowadays to identify the action recognition, which does not require any information about human body or movement of people. It is divided into 2 parts local Spatio-temporal interest points and trajectory detector. We are using STIP so that we will discuss this method further.

Laptev and Lindeberg are the people who worked on local feature detectors on video. [21]They utilized Harris3D interest point detector, which is the further form of the Harris detector to space and time domain by utilizing video values in space-time

to provide critical variations in both dimensions.[21] To search the eigenvalues of the matrix of any video, Harris3D is used that calculates a Spatio-temporal second-moment matrix at each video points. The final points are the local positive Spatio-temporal maxima

Dollar sometimes observes right spatiotemporal corners are unable to detect an interesting motion; for example, human face expression that is why Gabor detector is used. Gabor detectors give much denser results than Harris3D, and it applies a set of Gaussian kernels and filters. [21]

Willems proposed that the Hessian matrix is calculated by Hessian3D detector and uses the determinant of the Hessian matrix for each scale selection and point localization.[22] Detectors use internal video to speed up calculation by taking approximate derivatives with box-filter. [22] These detectors are dense and scale-invariant if we compare with Harris3D, but they are not as denser as Gabor detector.

Dense sampling is also another technique which is used to extract interest points at average position and scales in space and time. Dense sampling uses 5 dimensions (x,y,t, tau, sigma) where x, y and t are the Spatio-temporal position of the point in the video, sigma is the spatial value and tau is the temporal scale. These detectors extract a large number of features, and simultaneously, it also extracts the essential features of the video.

2.4 Spatio-Temporal Gesture Recognition

Hidden Markov Model is used for gesture recognition and skill assessment, which were based on robotic minimally invasive surgery. These models initially identify the gestures and motions of the specific surgical task and mostly used to identifies the surgical activity recognition and for skill assessment. These surgical activity recognition assessment based on methods like linear dynamically systems (LDS) and a bag of words(BOW) models.[23, 24] Nowadays, Deep learning or neural networks have been used for surgical tool detection and gesture recognition. [25, 26]

Bow is the state-of-the-art technique which doesnot capture the underlying structural information,derived from local spatio temporal features and used for video activity recognition [27] BoW overcomes this limitation by using short motion video sequence and encoded into temporal and structural information which is passed into

BoW model. It is reported that with A-BoW technique, accuracy is detected much higher than traditional BoW technique. [28]

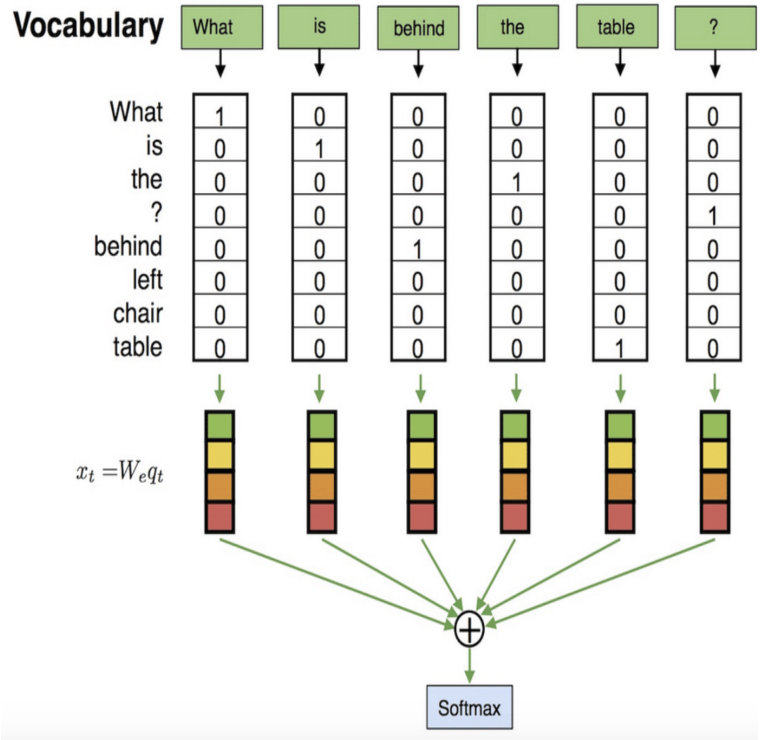

Figure 2.2: Bag of Words representation of the input that is next follow by 'Softmax' [2]

There are many other techniques which are used to analyze the time sequence data includes motion texture for a score prediction, which is encoded into frame kernel matrices and by texture analysis. Sequential Motion Texture (SMT) is used into motion technique which divides video time series into sequential time windows.[28]

Our work is different from RMIS works, which only automate the general surgical knot and don't use robotic kinematic data. We are using two classes that are forefinger throw, and backhand throw to complete a full knot for each novice, intermediate and expert surgeons and analysis the accuracy of the person with the help of STIP. [28]

2.5 Deep Learning

Deep learning is a part of machine learning, and it can be supervised or unsupervised. Nowadays deep learning architecture like CNN, ANN, deep neural networks applied to every single application for example board games, computer vision, image recognition, NLP and produces better results which are comparable to and in cases better than human beings [29]. Traditionally CNN model includes a series of convolution and pooling layers followed by hidden dense layers. Feature extraction is done by convolution and pooling, whereas MLP saw as a classifier for hidden layers. [30]

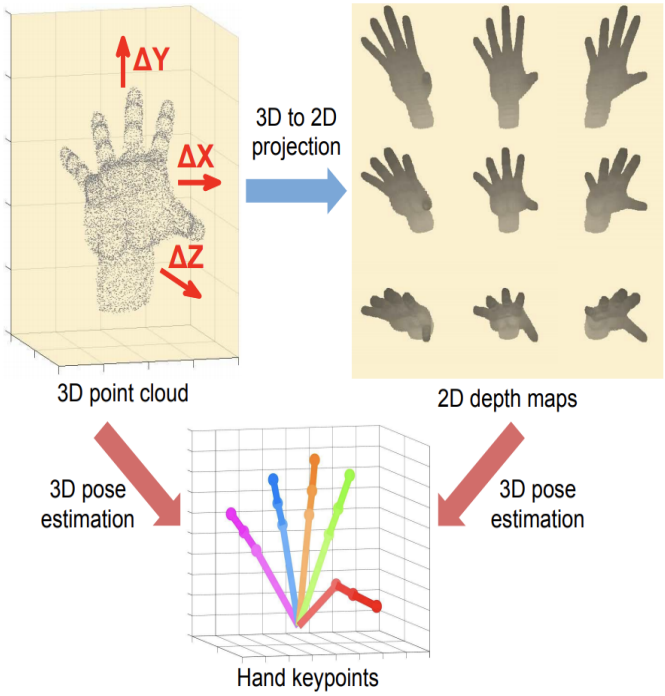

Figure 2.3: 2D depth image has many to one relation whereas 3D depth has one to one relation with 3D pose. [3]

Most of the hand techniques are divided into 3 methods generative, discriminative, and hybrid.[3] When pre-defined hand model is assumed and fit it into depth image by minimizing hand-crafted cost functions, then that method is known as generative whereas methods those are localized hand joints from an input map is known as discriminative.[3] These model also utilizes hand-crafted features which overcome

CNN approaches that learn features by themselves. [3]

A common framework for 3D human and hand pose estimation from a depth image is that it takes a 2D image and gives 3D coordinate like hand joints etc.[3] However, there are some consequences like there is some distortion in 2D depth image because the distance of the object points from depth camera is just physical (pixel value of the depth map represents the physical distance of objects from the depth camera) as most of the method takes a depth map of the image which distorts the shape of an actual object in the 3D form if we project them in 2D image space. Therefore the network cannot identify the image and performed distortion estimations.[3]

After all researches, Zheng found a general framework for different sequences classification called Multi-Channels Deep Convolution Neural Networks, or MC-DCNN. In this framework, the neural network consists of various 1D time-series sequences, and the features are executed individually. After that, each trained feature is merged and concatenated, which uses classic MLP and places at the end of the feature extraction layer. Because of skeleton structure agnostic, MC-DCNN made different from other deep learning (gesture recognition) models. [3]

Chapter 3

Literature Review

As the technology of Computer Vision increases, there is a high demand for hand gesture recognition ,detection and tracking in the market. While an expansive literature survey of this field would be a mind-boggling challenge in light of an enormous number of publications, we will concentrate in this section on a review of the most representative advancements that are suitable to my theory research topics.

To interact with the world, human hands plays a most improtant role in it. These hands are highly articulated structures with some 27 degrees of freedom (DOF) [31].This high DOF made hand gesture recognition a complicated task.Despite the fact that hand postures and motions are often considered as being indistinguishable, there are contrasts as explained in [32, 33].

This literature review is based on the paper "Video Based Assessment of OSATS Using Sequential Motion Textures" by Irfan Essa where he used STIPs for the feature detection and used SVM on those feature by clustering features and made a vocabulary of it.

3.1 Harris 3D

Harris Corner Detection was introduced by Chris Harris and Mike Stephens in their paper. It is combination of Corner and Edge Detection which was introduced in 1988 and known as Harris Corner Detection. He took simple mathematical formula which finds the difference in the intensity for a displacement of (u,v) in every direction shown

in equation [4]

$$E(u,v) = \sum_{x}, yw(x,y)[I(x+u,y+v) - I(x,y)]^2 \tag{3.1}$$

where w(x,y) is window function which is either rectangular window or Gaussian window gives weights to pixel , I(x+u,y+v) is shifted intensity and I(x,y) is the intensity.[4]

Maximize this function E(u,v) to detect corners mean maximize the second term and apply Taylor's expansion to above equation, we get[4]

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{3.2}$$

where M is ,

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \tag{3.3}$$

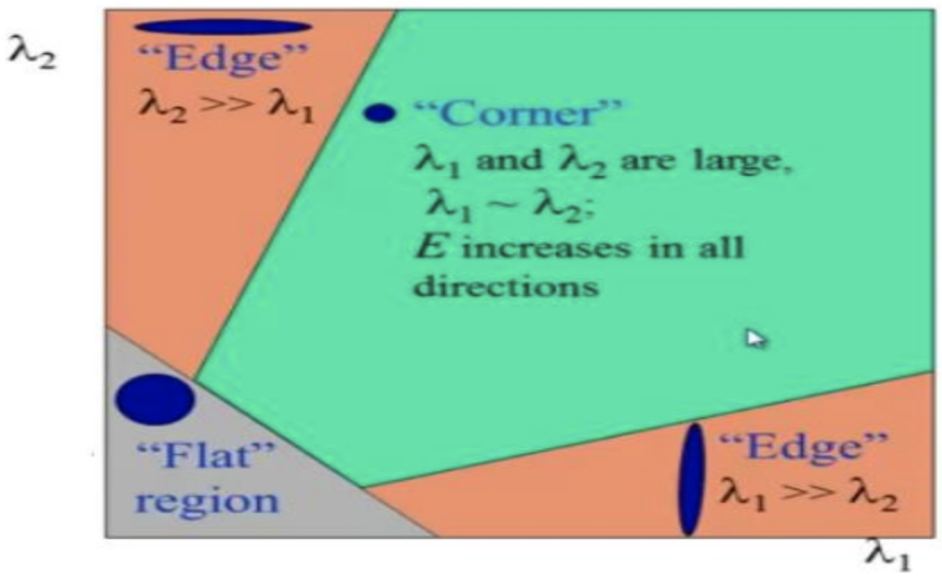

Figure 3.1: Harris Corner Detection [4]

Here, I_x and I_y are image derivatives in x and y directions respectively.

Now, they created a score which is also an equation which determines if a window have corner or not:

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{3.4}$$

where,λ_1 and λ_2 are the eigen values of M.

In this case, values of eigen will decide whether it is corner, edge or flat surface which is shown in Figure 3.1. [4]

1. If mod R is small, the area is flat because λ_1 and λ_2 are also small.
2. When R is less than 0 then the area is edge because $\lambda_1 >> \lambda_2$
3. When R is large, then area is corner which means λ_1 and λ_2 are also large in size and λ_1 almost equal to λ_2

3.2 Features Extraction Using STIP

Laptev extends the spatial domain into saptio-temporal domain by using space-time value features of the image in both dimensions. The point from this property will saptio interest point with different location and time of the moments non-constant motion of the image in a local Spatio-temporal neighbourhood.[34]

To design this sequence model, function is used $f : R^2 \times R \to R$ and $L : R^2 \times R \times R^2_+ \mapsto R$ is used to construct its linear scale space representation by convolution of f with an anisotropic Gaussian kernel with distinct spatial variance σ_l^2 and temporal variance τ_l^2 .[34]

$$L(\cdot;\sigma_l^2,\tau_l^2) = g(\cdot;\sigma_l^2,\tau_l^2)f(\cdot), \tag{3.5}$$

where the spatio-temporal separable Gaussian kernel is defined as

$$g(x,y,t;\sigma_l^2,\tau_l^2) = \frac{exp(-(x^2+y^2)/2\sigma_l^2 - t^2/2\tau_l^2)}{\sqrt{(2\pi)^3\sigma_l^4\tau_l^2}}[34] \tag{3.6}$$

The code extracts STIPs (densely sample interest points) and finds out the local space and time descriptors. This implementation is somewhat based on the extended

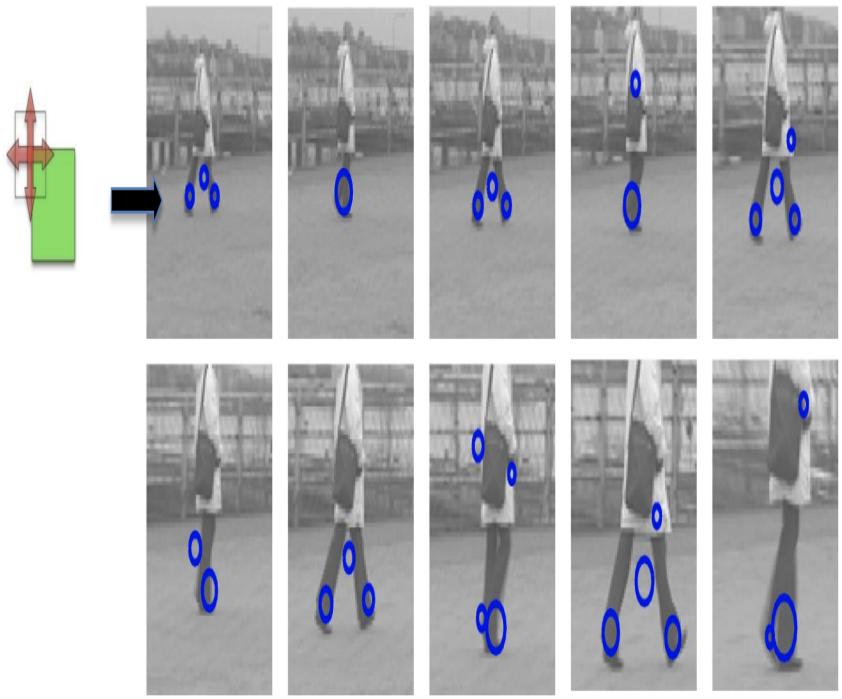

Figure 3.2: Features Extraction Using STIP

space-time Harris Detector. Scale selection is not implemented by the code but gives the points for a bunch of many combinations of space and time scales. This description gives better results in many application like walking, running and many more in hand gesture activity.[35]

Descriptors for spatio-temporal patches

1. At each spatio-temporal interest point, descriptors are characterized considering the volume of the cuboid neighbourhood. The size of the cuboid is obtained from the scale as (k σ) (kσ) (k'^{τ}) with k is equal to 6.

2. The common framework for the Descriptors of the volume are [5]:

   (a) **Preprocessing:** Gaussian 3D kernel smoothed all volumes

   (b) **Spatio-temporal pooling:** the volume is sub-partitioned into various smaller cuboid volumes(e.g. 3x3x2 cuboids).

(c) For each pixel,a function is executed to get invariance to enlightenment and rotation(Feature calculation) which is trailed by feature quantization(histograms of the computed features are accumulated) as shown in figure 3.2:

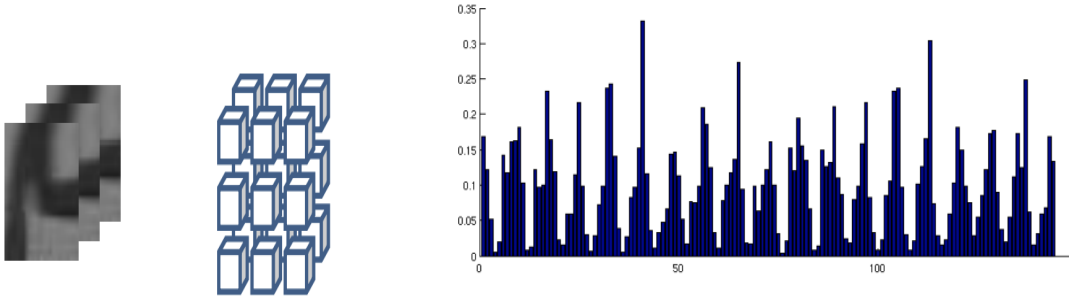

Figure 3.3: Descriptors for spatio-temporal patches [5]

This code detects the descriptors like HOF (Histograms of Optical Flow) and HOG (Histograms of Oriented Gradients), which is computed on the video patch in the area of each identified STIP. The patch is divided into grid of 3x3x2 ST blocks where 4-bin HOG and 5-bin HOF are computed for each blocks and merged into 72 elements and 90 elements descriptor respectively. [35]

For each sample video (e.g. bht1.avi), the code will take a text file as input argument (e.g. bht1.txt) to compute HOG/HOF descriptors which is the following format:

point-type, x, y, t, sigma2, tau2, detector-confidence (hog/hof)

where "point-type" can be any integer value, and "sigma2" and "tau2" are the spatial and temporal scale values respectively. The following scale values are expected:[35]

sigma2 = {4, 8, 16, 32, 64, 128, 256, 512}, tau2 = {2, 4}

The will code quantizes the input spatial/temporal scale values to the ones above in case of a mismatch. sigma2 and tau2 define the spatial and temporal patch sizes for the descriptor around feature locations as [35]

spatial patch size = 2 * szf * sqrt(sigma2) temporal patch size = 2 * tszf * sqrt(tau2)

where "szf" (default 9) and "tszf" (default 4) are the spatial and temporal patch size factors respectively.[35]

3.2.1 Histograms of Oriented Gradients

In Computer Vision and image/video processing, HOG is the feature which is used to detect objects. It counts the occurrence of gradient orientation in video or image in part of the image like ROI (Region of Interest) [6]

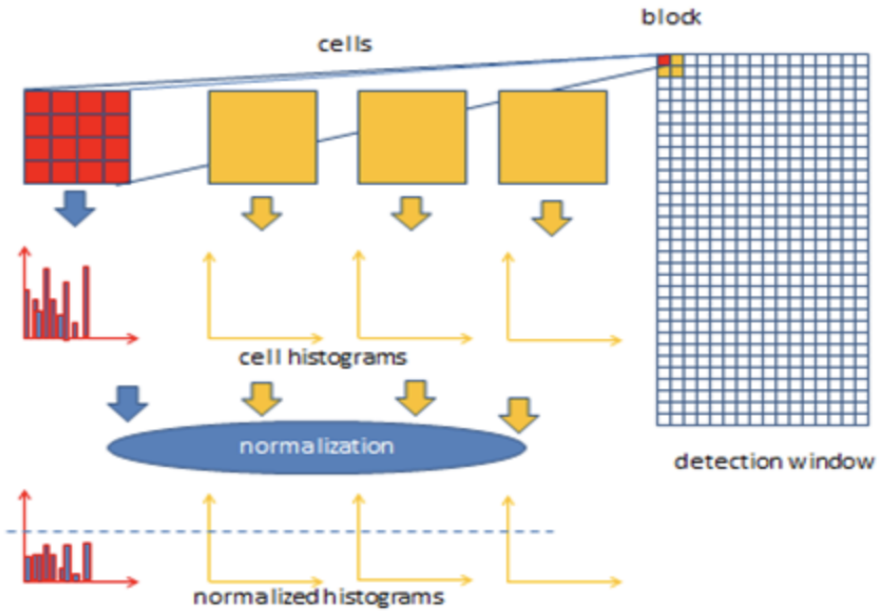

Figure 3.4: Demonstration of the HOG Algorithm [6]

1. For videos, each frame is considered as image and for that divide each frame into small connected regions which is known as cells and for each cell compute the histogram of gradient directions for a pixel within the cell.[6]
2. Divide every cell into angular bins, which is based on gradient orientation.[6]
3. Each cell's pixel contributes as a weighted gradient.
4. Block is a group of adjacent cells which is considered as a spatial region. The grouping of the cell is the base of grouping and normalization of histograms.[6]
5. Normalized group represents the block histogram, and further, the set of these blocks is known as a descriptor.[6]

3D Gradient

To calculate 3D Gradient at every pixel, differentiate the image function $I(x, y, t) \rightarrow R$ which will give 3 channels(3.3, 3.4, 3.5) [5]:

$$G_x(x, y, t) = I(x + 1, y, t) - I(x - 1, y, t)[5] \tag{3.7}$$

$$G_y(x, y, t) = I(x, y + 1, t) - I(x, y + 1, t)[5] \tag{3.8}$$

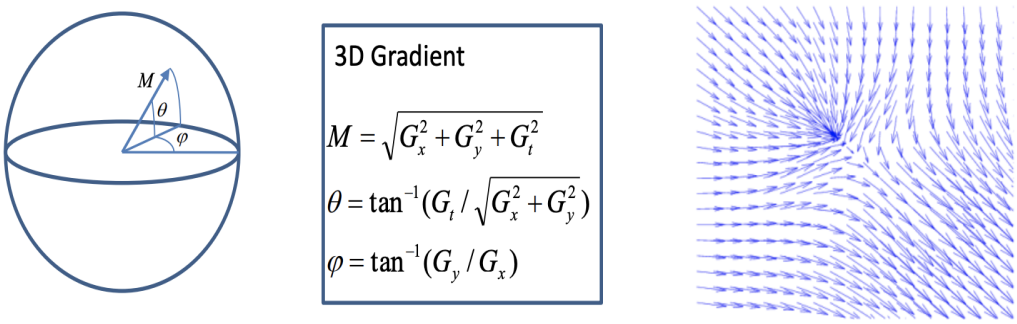

Figure 3.5: Magnitude and Orientation [5]

$$G_t(x, y, t) = I(x, y, t + 1) - I(x, y, t - 1)[5] \tag{3.9}$$

Here, the gradient is represented as magnitude M, and Orientation is represented by θ and ϕ shown in Figure 3.4.

3.2.2 Histograms of Optical Flow

Optical flow is the most popular thing in CV where it shows the pattern of motion of objects, surface in a video which is caused by relative motion between an observer and an object [7]. In other words, it will measure the motion of the pixel of two frames to time. It was presented by American therapist James J. Gibson during the 1940s to evaluate visual view stimulus gave to creatures moving through the world.[7]

The method of optical flow tries to calculate the motion between two frames, which is taken at different time stamps t and t + Δt in 3D dimension. These methods are

differential as partial derivatives are used to space and time coordinates.

If we take 2D+ t dimensional case at location (x,y,t) with intensity I(x,y,t) and move by Δx, Δy and Δt between two frames then the intensity will be [7]:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t)[7] \tag{3.10}$$

And if we assume the movement is too small then the image constraint at I(x,y,t) will be :

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\delta I}{\delta x}\Delta x + \frac{\delta I}{\delta y}\Delta y + \frac{\delta I}{\delta t}\Delta t + H.O.T.[7] \tag{3.11}$$

from these we get,

$$\frac{\delta I}{\delta x}\Delta x + \frac{\delta I}{\delta y}\Delta y + \frac{\delta I}{\delta t}\Delta t = 0.[7] \tag{3.12}$$

or

$$\frac{\delta I}{\delta x}\frac{\Delta x}{\Delta t} + \frac{\delta I}{\delta y}\frac{\Delta y}{\Delta t} + \frac{\delta I}{\delta t}\frac{\Delta t}{\Delta t} = 0[7] \tag{3.13}$$

which will give

$$\frac{\delta I}{\delta x}V_x + \frac{\delta I}{\delta y}V_y + \frac{\delta I}{\delta t} = 0[7] \tag{3.14}$$

where V_x , V_y are x and y components of the velocity, $\frac{\delta I}{\delta x}$, $\frac{\delta I}{\delta y}$ and $\frac{\delta I}{\delta t}$ are the derivatives of the image at (x,y,t) in corresponding directions. [7]

Thus,

$I_x V_x + I_y V_y = -I_t$

3.3 K-Means Clustering

There are many unsupervised Machine Learning algorithms just like K-Means clustering, and it will find the fixed number of clusters or group (k) in unlabeled data. It means it will form a cluster which is a group of data points that are in a group of same

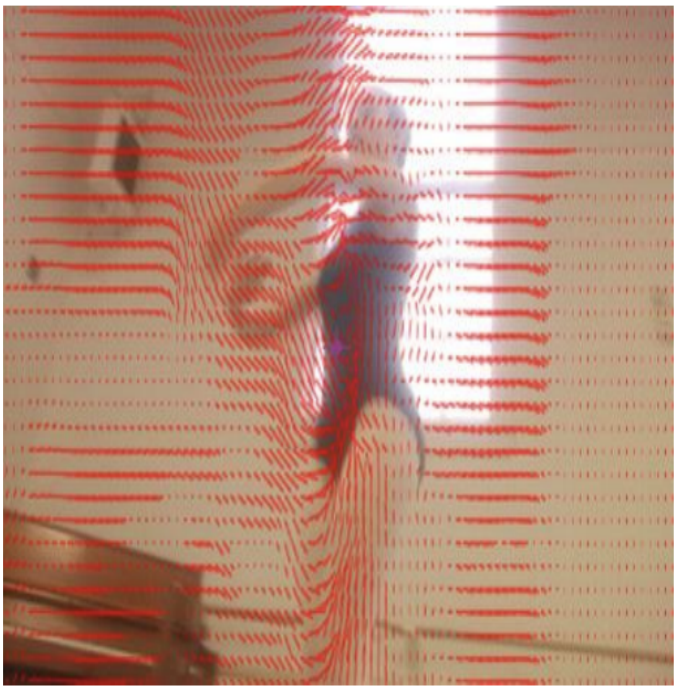

Figure 3.6: HOF representation [7]

feature kind. A centre (centroid) is defined when we use K-Means algorithm. Every point is a part of the cluster whose centre is nearby to keep centroids small.[36]

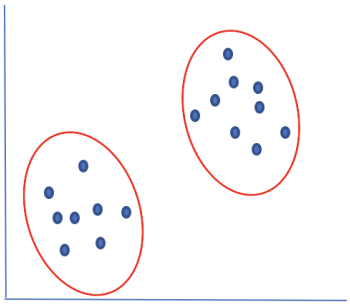

Figure 3.7: K-mean Clustering

Algorithm

1. First of all, divide the cluster of data into k parts where k is already defined.
2. Now, select k points as cluster centres.
3. According to Euclidean distance, find the distance between points and centre and assign to that centre.
4. Find out the mean or centroid in each cluster.
5. In the end, repeat all steps 2, 3, 4 until the same point is assigned to each cluster in every second round.

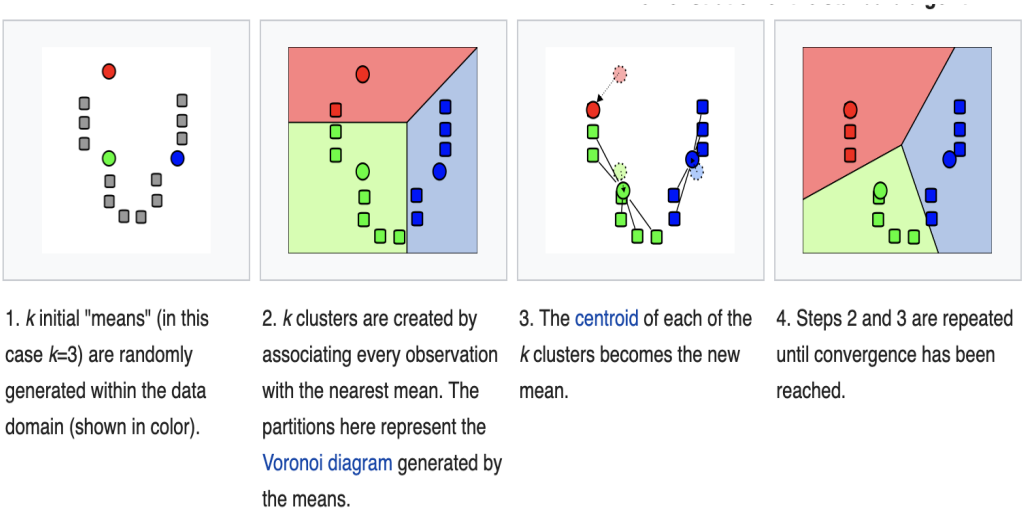

Figure 3.8: Demonstration of the standard algorithm [6]

3.4 Bag-of-Words

Bag-of-word is the most common technique used in CV(Computer Vision) in the field of image and video classification. In the bag of words, we count the number of each word appear in the document, use its frequency to know about keywords of the document and make a frequency histogram from it. [8]

Figure 3.9: Bag-of-visual-words (BOVW) [8]

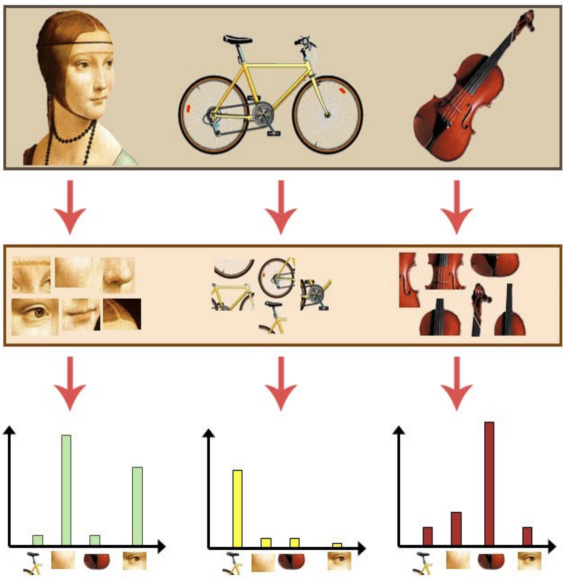

Figure 3.10: Histogram representation of images[8]

It is like a document full of words, but in the document, there are words whereas, in videos and images, we use image features as the words.

Features consist of descriptor and key-points. If we take a picture, its key-points are always the same no matter if the image or video is in an inverted position, shrink or expand.

3.5 SVM

Vladimir N. Vapnik and Alexey Ya invented Support Vector Machine. Chervonenkis in 1963. It is the most popular linear classifier. The sample data used for SVM must

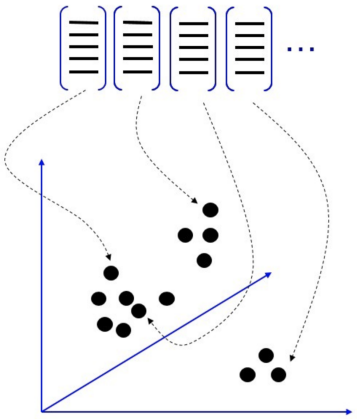

Figure 3.11: Feature Extraction by k means [8]

be labelled; that is why it is known as supervised learning models.[9] SVMs find a hyperplane(or a line in any dimensions greater than 2) in between the dataset such that the distance between the next-closest data on both sides is maximized. In other words, it will calculate a boundary of maximum margin that leads to a homogeneous partition and known as maximum margin classifier. This algorithm can also be used for regression. [9]

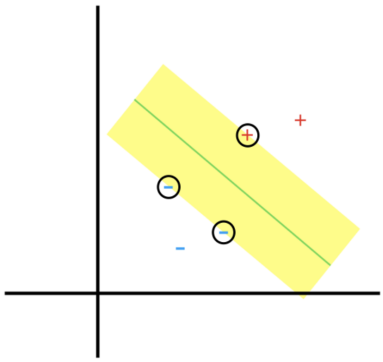

Figure 3.12: Hyperplane of Support Vector Machine[9]

To separate two classes from each other equation 1 and 2 represents the formulas for a line or hyperplane(represented equation 4.3 and 4.4). For all the data points, an SVM should find the weights and separated according to the decision rule. [9]

$$\overrightarrow{w}.\overrightarrow{x} + b = 0 \tag{3.15}$$

$$y = mx + b \tag{3.16}$$

If we take an example of two sets which is negative and positive values in 2D space, with a straight line between two classes of data points. The yellow space in between the two classes shows the margin between the points that are close to each other and the circle points are the support vectors. The hyperplane is considered as n minus 1-dimensional subspace for example if we consider or space as 2D then its hyperplane will be 1D which is just a line, and for 3D space, the line is converted into the plane in 2D dimension, equation 4.5 represent hyperplane

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + ... + \beta_n * x_n = 0 \tag{3.17}$$

moreover, if this equation is greater than 0, then the dots are above this line and it the equation is less than 0 then dots are below this line.

SVM is trying to find the hyperplane with maximum margin in the linearly separable case with one condition that both the classes are classified correctly. If we see our dataset or any other dataset, then it is not probably linearly separable, and in that case, hyperplane will never meet this condition. So for non- linearly separable data, there are two concepts: Soft margin and kernels tricks.

1. In the soft margin, it will try to find a line which can accept one or two misclassified dots or points.

2. In kernels trick, it will try to find non-linear decision boundary.

3.5.1 Soft Margin

In the soft margin, there are two kinds of cases, which are as follows:

1. In the first case, when the point is on an inappropriate side of the decision boundary yet the margin. [9]

2. In the second case, the point is on the inappropriate side of the decision boundary as well as the wrong side of the margin. [9]

To find the correct decision boundary, we need to provide the tolerance factor. In Sklearn, it is represented as **C** which is known as penalty term, the bigger the C, the more penalty SVM gets. Therefore, the narrower the margin is, fewer the support vector will depend on. [9]

3.5.2 Kernel Trick

This is the most interesting trick, and it utilizes the existing features, perform some transformation and create new features. Those features are the key to find the non-linear decision boundary. [9]

In Sklearn, there is linear, rbf, sigmoid, precomputed or callable type of kernels we can choose.

1. For the **Polynomial Kernel**, it will transform some of the datasets from the existing features which help to create hyperplane. [9] If we take an example of some dataset x, and we transform that dataset by x square, then it will give parabola, and we can easily segregate the data points and create hyperplane of it. [9]

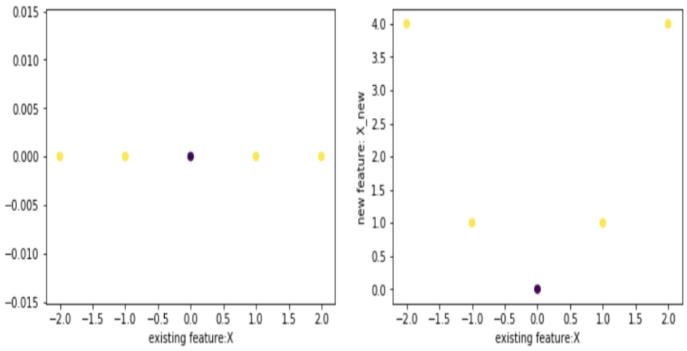

Figure 3.13: Soft Margin [9]

2. **Radial basis Function** is popularly knowns are rbf, it will transform dataset by measuring the distance between all other dots to the specific dot centre. [9]

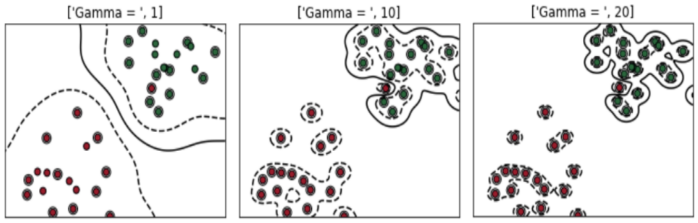

Figure 3.14: Gamma values for rbf [9]

The most basic RBF function is Gaussian RBF, which is represented by equation 4.6. [9]

$$\phi(x, cneter) = exp(-\gamma||x - center||^2) \tag{3.18}$$

Here, gamma controls the influence of new features on the decision boundary. [9]

Chapter 4

Design

This section is all detailed analysis of surgical knot tying data collected with the help of people.

4.1 Overview of the Pipeline

The primary purpose of this thesis is to find out the accuracy of tying a one-handed surgical knot of two classes novice and experts. Based on background research, the following steps are implemented in this pipeline.

1. First of all, we collect all the video data of different people performing knot tying, including surgeons and novice. Then we segregate all videos according to the right and left hand.
2. Next step is inserting these video file to Software STIP which was created by Laptev.
3. In the next step, identify the clusters by using k-means clustering.
4. Final step is to apply SVM to calculate the accuracy of the classes.

4.2 DataSets

Input of the system is the video of surgical knot where 9 participants participated in the knot tying process. The dataset was divided into two parts from the camera

prospective (the camera position is at 90° of angle, and an angle of 60° takes another dataset where the camera is head-mounted, and movements are correctly captured) and divided also into two class (fft and bht)

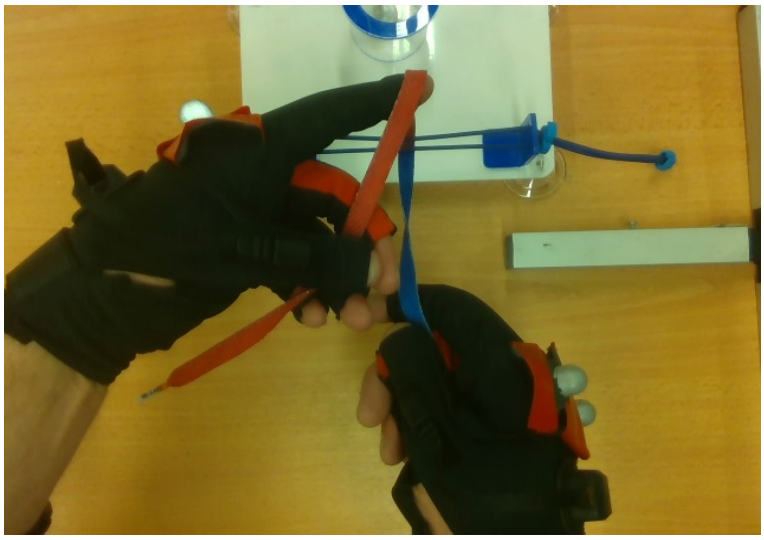

Figure 4.1: This image is based on the camera position is at 90°, vertically mounted on the stand

In some of the cases, data is captured by using hand detecting sensor gloves to capture the motion of fingers. As we can see in figure 3.1, hand gloves are used to capture motion data.

The project dataset contains 110 video files that capture 56 fft and 55 bht of 9 people, including 60° and 90° angle position.

4.3 Hardware -Real Sense Camera

In the world of technology, there are many different machines which can able to see things which are not visible to human eyes. Depth sensor camera is also known as time-of-flight camera, which can resolve the distance between object and camera. Today the market is full of 3D depth image camera in low price with better quality.

However, the first structured light camera is first-generation Kinect, which was launched in 2010, and the second-generation camera is a time-of-flight launched in 2013 by Microsoft. With the invention of the first generation depth camera, scientist started

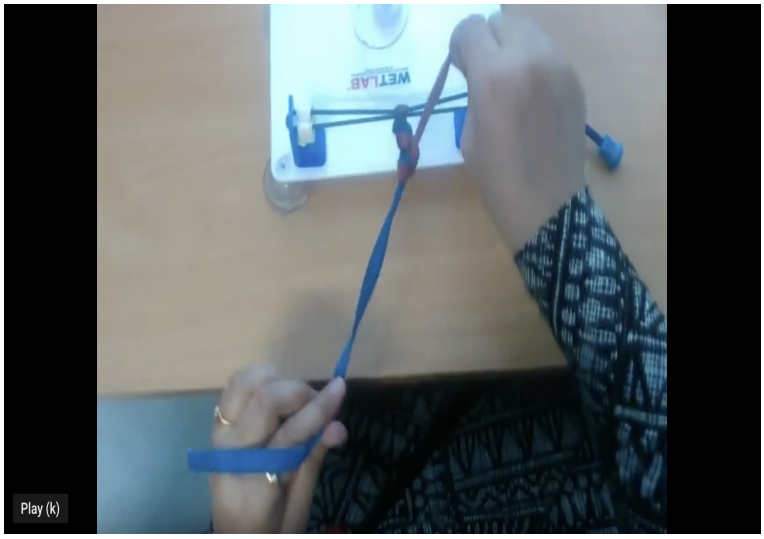

Figure 4.2: This image is based on the camera position is at approx 60° which is egocentric

Figure 4.3: Intel RealSense d435

using in computer vision programs which can identified skeletal and face tracking, but not able to distinguish the fingers on a hand. In this dissertation, we are using the RealSense D435 camera invented by Intel. D400 series was launched in 2018 with Vision Processor D4. One of the features is high depth resolution, which is up to 1280x720 at 30 frames per second (fps) and broader view. This camera provides accurate depth

perception when objects are moving, or in motion, it also covers more view of the area minimizing blind spots [37].

4.4 Software - DepthSense SDK

For implementing the project, Intel real sense software (SDK) used, which allows depth and colour streaming and all information. Its library also offers for builtin session to record in .bag format and playback of streamed course.

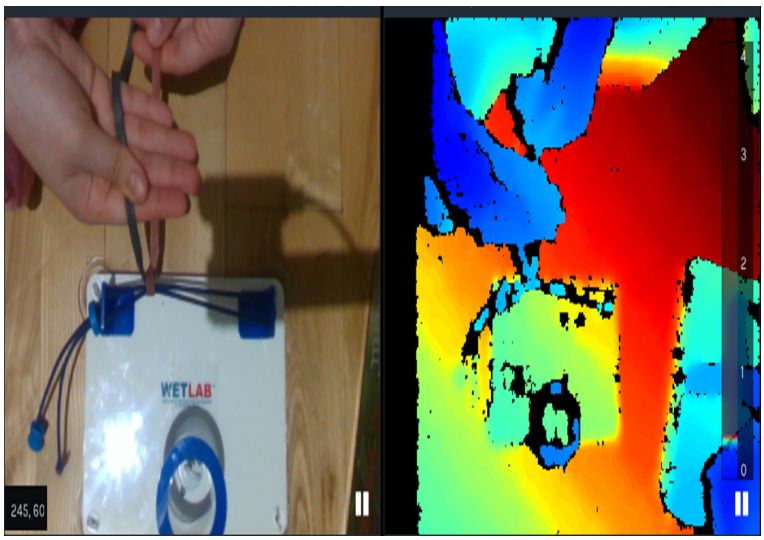

Figure 4.4: Output image of DepthSense SDK

Chapter 5

Methodology

The proposed surgical knot tying system consists of two stages: extracting features from STIP code and Train an SVM model. There are 3 vital factors which affect the accuracy of the system: first is the quality of videos, the number of video samples and the number of k means clusters to build the model. The following sections are about the process followed in the project.

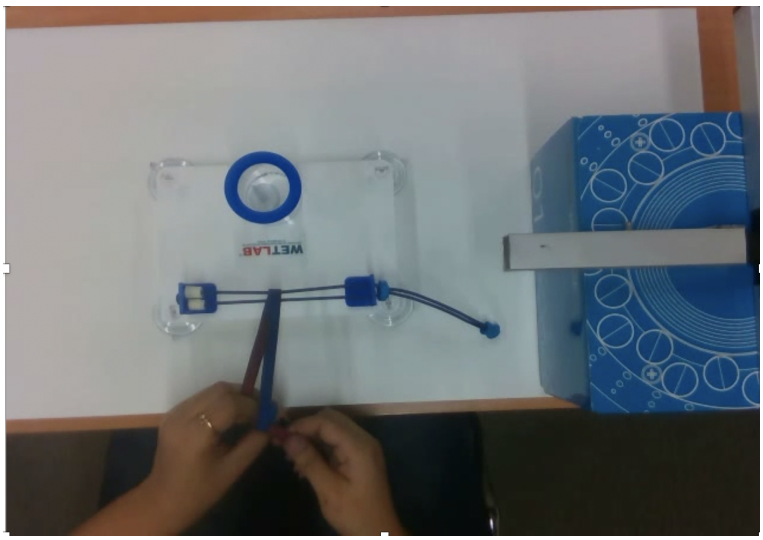

Figure 5.1: FFT(Forefinger throw)

The video of surgical knot is divided into two parts FFT(Forefinger throw) Figure 5.1 and BHT(Backhand throw) Figure 5.2 and features like Harris Corner, HOG and HOF are extracted on those video as shown in Figure 5.3.

Figure 5.2: BHT(Backhand throw)

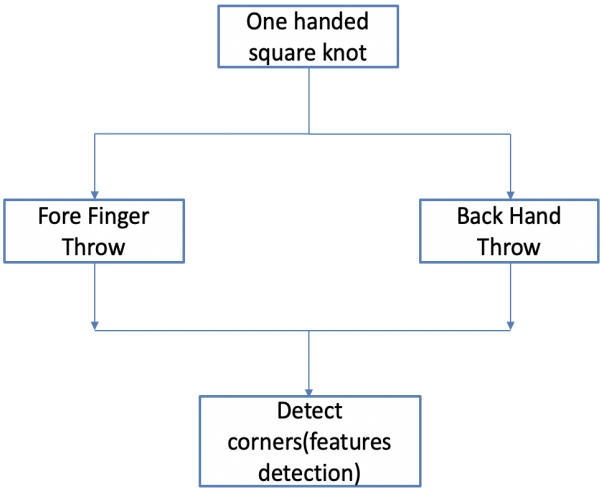

Figure 5.3: First Stage

For the first part of the stage, all the videos are divided into frames, and for each frame, Harris Corner Detection is applied on it which extracted every corner in the frame shown in Figure 5.4.

The bounding box is used to create a boundary line around the hands, so that extra irrelevant features are deduced in this process but did not achieve any good results. Expect hands all other corner deducted in this process.

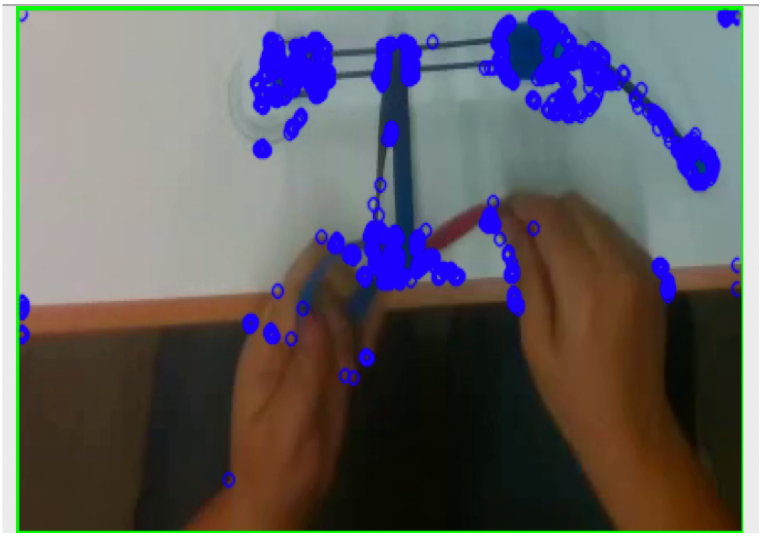

Figure 5.4: Harris Corner on frame with bounding box

Next approach is to use Harris Corner on the video to find out the better results from the previous approach, as the position of hand changes in the video Harris Corner failed to detect the hands again but the results(shown in Figure 5.5) from the previous result was better this time. Still cannot use for further extraction.

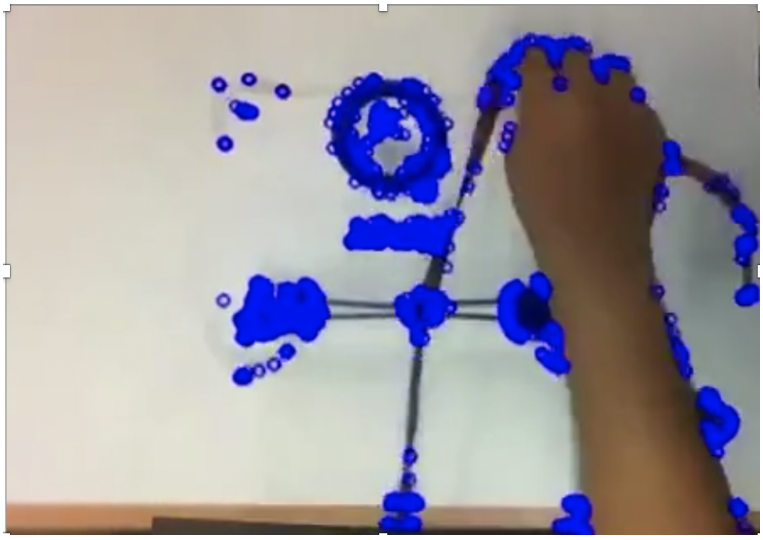

Figure 5.5: Harris Corner on frame

Here, the number of the interest points reduced as compare to frame approach

but still not efficient to extract the correct features. From the analysis of Laptev's approach in 2005, extraction of the feature of moving object in the video, we extracted the Spatio-temporal interest points (STIPs) from it. STIPS is the extended version of spatio domain (STIF) to the temporal domain. These Spatio-temporal corners exist in the district of the high variety of image intensity in each of the three directions (x, y, t). This requires that Spatio-temporal corners are located at spatial corners such that they invert motion in two consecutive frames (high temporal gradient variation) [5]. They are identified from local maxima of a cornerness function computed for all pixels across spatial and temporal scales, as shown in Figure 5.6. [5]

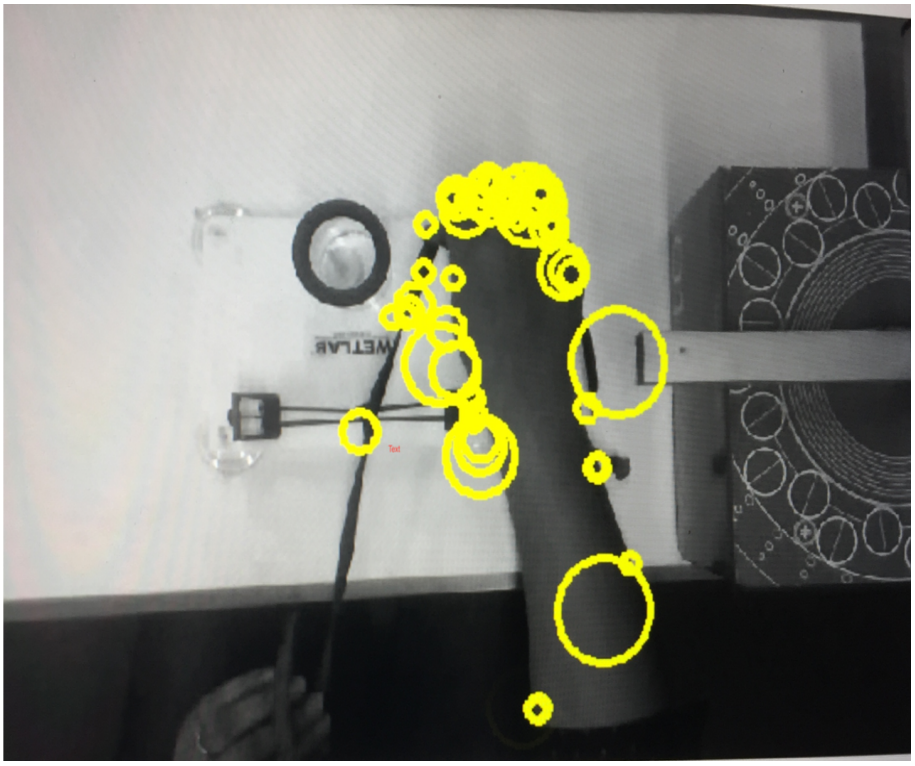

Figure 5.6: STIPs on sample video

To get the frame kernel matrix, we identify the Spatio-temporal interest points (STIP). We process the HOF (Histogram of Optical Flow) and HOG (Histogram of Oriented Gradients) on a 3D video patch around each distinguished STIP to get a 162 component HOG-HOF descriptor for every frame interval from 0 to 100000000. By default, the spatial patch size factor is 9, and temporal patch size factor is 4.

The patch is parcelled into a grid with 3x3x2 Spatio-temporal squares; 4-bin HOG descriptors and 5-bin HOF descriptors are then figured for all blocks and are linked into a 72-component and 90-component descriptors separately. Figure 5.7 shows the movement of a dancing girl divided into 3x3x2 Spatio-temporal blocks and 4-bin HOG descriptors and 5-bin HOF descriptors are then computed for every block.

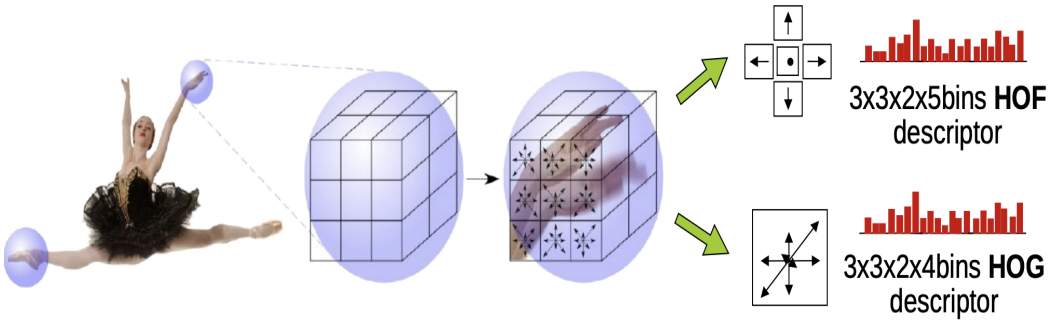

Figure 5.7: 4-bin HOG descriptors and 5-bin HOF descriptors

To compute the HOG/HOF descriptors for user-defined locations in the video, the code will take a text file as input argument (e.g. video-sample.txt)

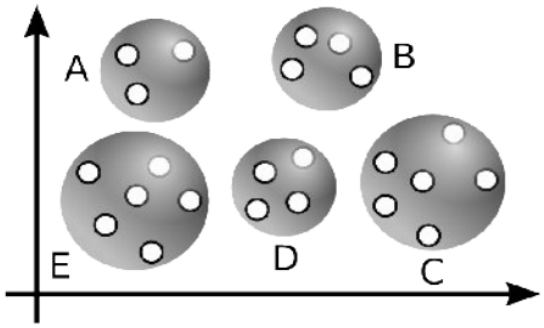

Figure 5.8: Example of k-means on dataset

for each video sample (e.g. fft.avi), with points in the following format in text format (e.g. fft.txt) with following columns like point-type, x, y, t, sigma2, tau2, detector-confidence (72-element HOG and 90-element HOF).

From the text file, we get a feature vector of HOG-HOF, classified each txt file to k distinct clusters by applying k-means clustering shown in figure 5.8. Here, k = 20

which is a random selection, if k¡20 then the clusters overlap and the accuracy was not good as expected. This partitions the entire frame into clusters where each cluster of represents a distribution for a specific movement class in the data. Each feature vector is allocated to its closet centre.

A whole video sequence is spoken to as event histogram of visual words which is otherwise called BoW (Bag of Words) shown in Figure 5.9.

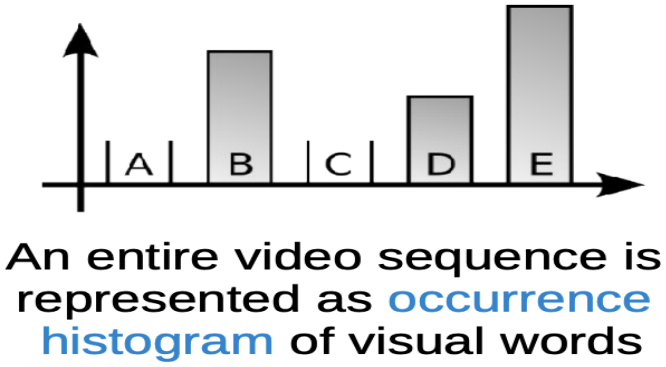

Figure 5.9: BoWs representation in the form of histogram

All the video file's feature vector is grouped by k-means cluster and then to check the occurrence of the motion class, a histogram of visual words were created. Finally, got a 1D feature vector which can be passed into linear SVM. Two classes, FFT and BHT, are labelled as 1 and 0 respectively. Read the data from the files fft.txt and bht.txt. These files have a 1D vector and label. The data is then part into training (80%) and test (20%) sets.

Next, made an instance of the Linear SVM classifier from scikit-learn. In order to get results,different kernels(Linear, RBF and poly) with different C and gamma values were used.Train a Linear SVM.[38] As it, depending on the information, the line isolates the two classes. This is done utilizing the fit technique for the SVM class. Parameter Tuning using GridSearchCV is used to get more accuracy. [38]

The module sklearn can do grid Search over parameters model selection. By specifying the parameters and vary them according to need will affect results. Here, parameters C and gamma are various means for each combination of C and gamma is attempted, and the best one is picked based by using clf.best estimator. GridSearchCV performs 3-fold cross-validation, which means the information is separated into three

sections and uses two sections for training, and one section for deciding accuracy. [38] This is completed multiple times, so every one of the three sections is in the training set twice and the validation set once. The accuracy for a given C and gamma is the average accuracy during 3-fold cross-approval.[38]

Chapter 6

Results

This section relates the outcomes acquired for the experiments performed in the previous chapter.

6.0.1 Experiment

These experiments were performed on the MacBook Pro with graphics card-Intel Iris Plus Graphics 640 1536 MB having memory 8 GB 2133 MHz LPDDR3 and Virtual Box Ubuntu with 4 GB base memory with graphic card- VMSVGA.

6.0.2 Results

The data (text file- bbt.txt)from the STIP code contains 169 elements of combine HOG-HOF with other points (x,y,t). Extracted HOG-HOF from it and deduced to small feature vector by using k= 20 in k-means clustering.

All the points in th video divided into clusters, each cluster of points represents a distribution for a particular motion class in the data. Then, each feature vector is assigned to its closest cluster centre. This will bring high dimension feature vector into smaller size. After that, with the help of BoW the entire video is represented as occurrence histogram of visual words. This again reduced the size of the feature vector into 1D vector. When SVM is applied, the dataset is divided into 80:20 ratio so that, 80% of the data used as training purpose and remaining 20% is used for testing purpose.

The correlation of the data after distribution is shown by the Figure 6.1.

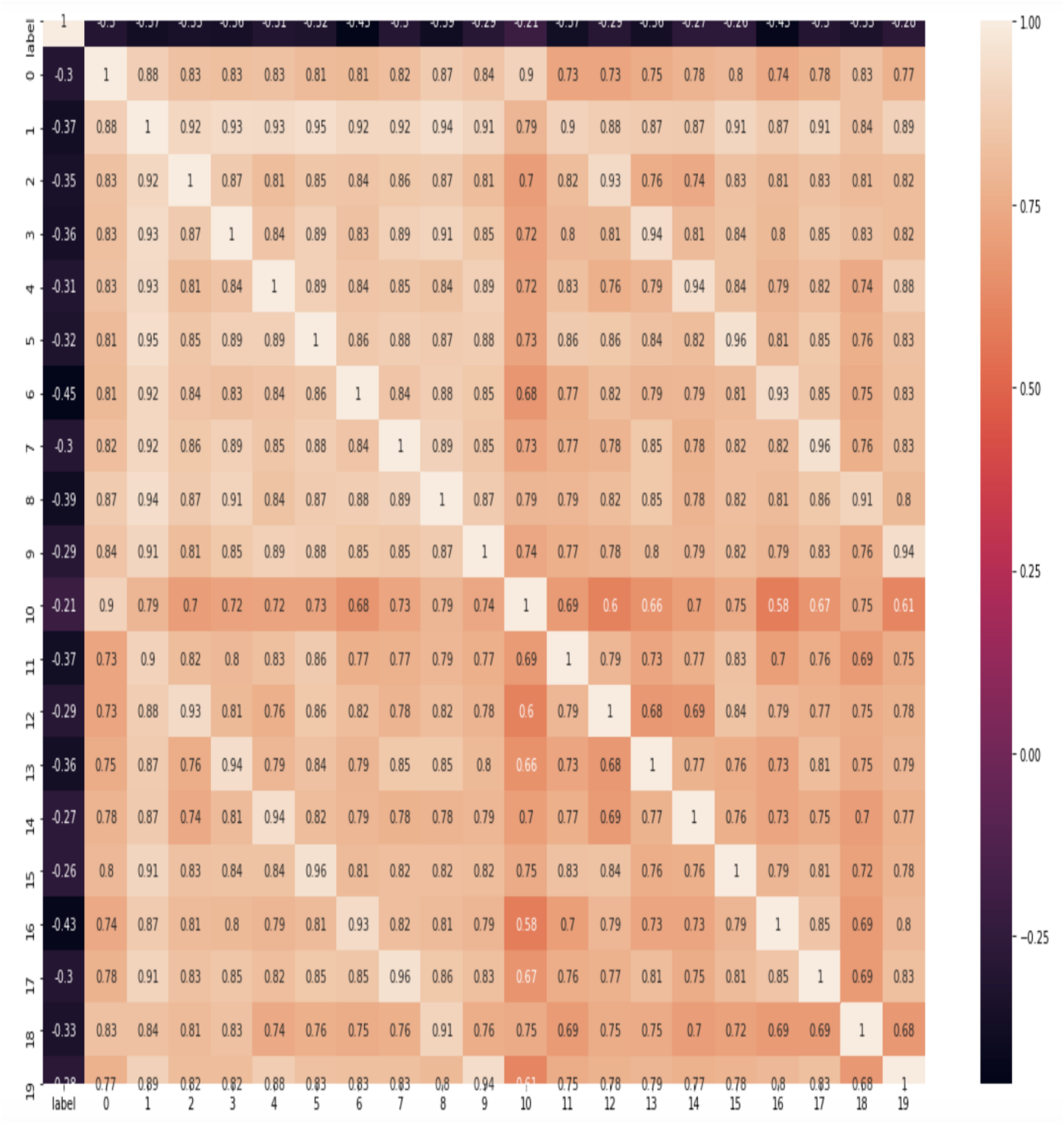

Figure 6.1: correlation of the data

6.0.3 Linear Kernel

When SVM is applied on the dataset with kernel= linear then, the results are much better than any other kernel with C= 1 and gamma = 'auto'. The confusion matrix for the linear is shown in Figure 6.2 where true positive = 4, false positive= 3, false negative= 1, true negative= 9.

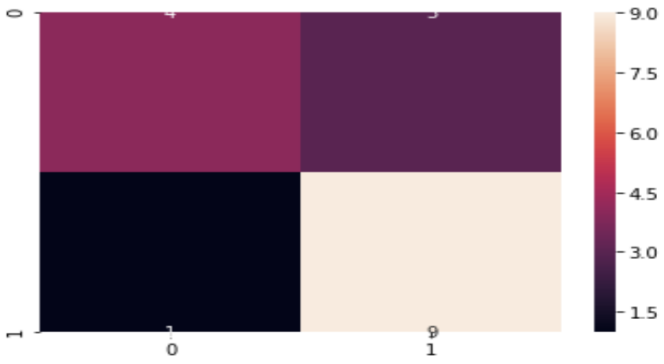

Figure 6.2: Confusion Matrix with linear kernel

And, the precision-recall value of the linear kernel is 73% whereas the accuracy is 76%.

```
Text(0.5, 1.0, '2-class Precision-Recall curve: AP=0.73')
```

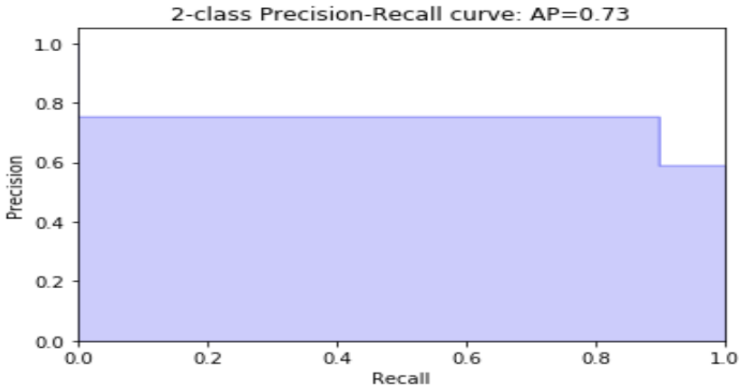

Figure 6.3: precision-recall value curve for linear

6.0.4 rbf Kernel

When SVM is applied on the dataset with kernel= rbf then, the results are worse if compare with any other kernel with C= 1 and gamma = 'auto'. The confusion matrix for the linear is shown in Figure 6.2 where true positive = 7, false positive= 0, false negative= 10, true negative= 0.

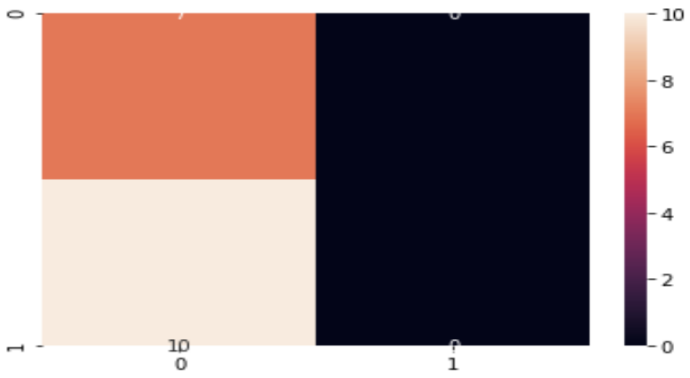

Figure 6.4: Confusion Matrix with rbf kernel

And, the precision-recall value of the linear kernel is 59% whereas the accuracy is 41%.

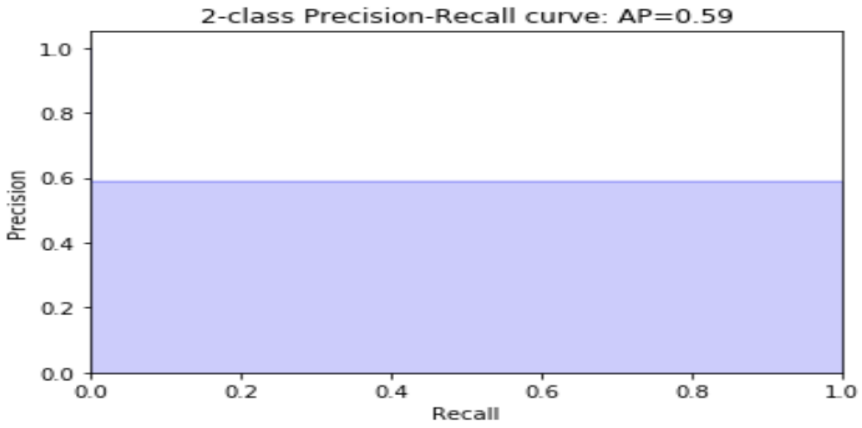

Figure 6.5: precision-recall value curve for rbf

6.0.5 poly Kernel

When SVM is applied on the dataset with kernel= rbf then, the results are worse if compare with any other kernel with C= 1 and gamma = 'auto'. The confusion matrix for the linear is shown in Figure 6.2 where true positive = 3, false positive= 4, false negative= 1, true negative= 9.

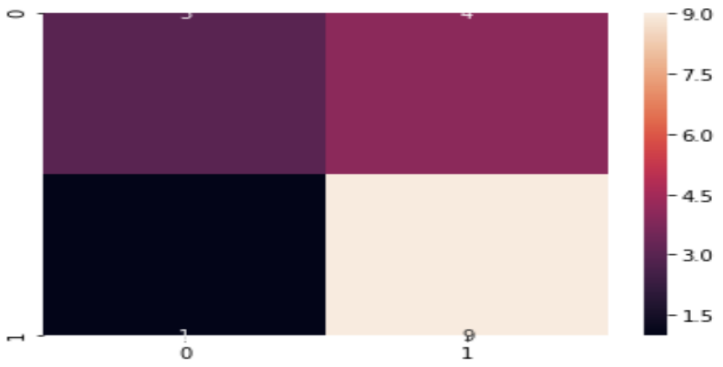

Figure 6.6: Confusion Matrix with poly kernel

And, the precision-recall value of the linear kernel is 68% whereas the accuracy is 71%.

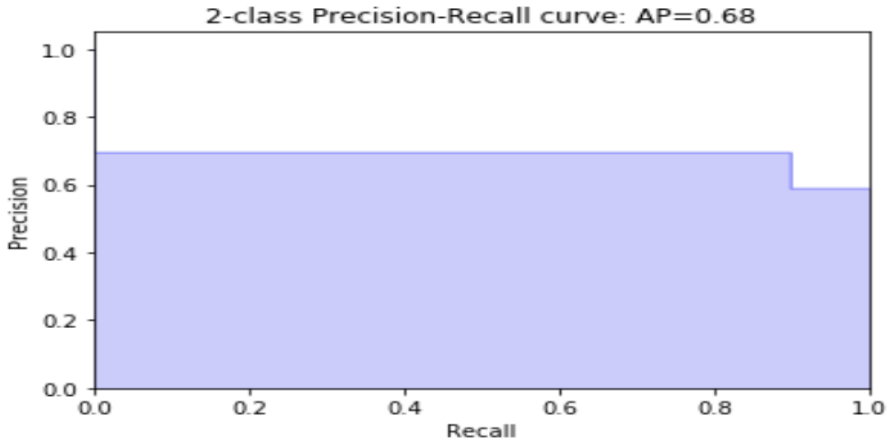

Figure 6.7: precision-recall value curve for poly

Table 6.1 shows the comparison between the ROC Score of the kernels used in the experiments.

Kernel	Score
Linear	0.73
rbf	0.51
Poly	0.68

Table 6.1: ROC Curve score for fft and bht

Kernel	Accuracy
Linear	76%
rbf	41%
Poly	71%

Table 6.2: Accuracy for 90 degree camera position

whereas, Table 6.2 shows the accuracy of the knot with two classes in SVM.

When GridSearchCV applied to the dataset, it divided the data into 3 folds means two part of the data is for training and other is for testing, this was done 3 times to get the better results and here with C= 1 and gamma = 0.00001, the accuracy is 71% and kernel used in this case is rbf. The confusion matrix is shown in Figure 6.8 where true positive = 3, false positive= 4, false negative= 1, true negative= 9.

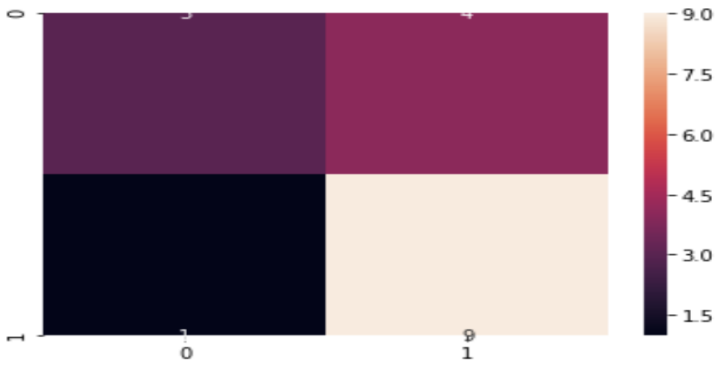

Figure 6.8: Confusion Matrix with 3-fold

Kernel	Accuracy
Linear	67%
rbf	17%
Poly	67%

Table 6.3: Accuracy for egocentric data

For the egocentric data, there were only 15 videos dataset for each class,so the accuracy linear and poly kernel were same and for rbf it was worse with 17% shown in the table 6.3.

Chapter 7

Conclusion

One of the main objectives of the thesis is to get the analysis of knots by using classical method of the Machine Learning which is based on the paper "Video Based Assessment of OSATS Using Sequential Motion Textures" by Irfan Essa. [39]

In the paper, he used the five techniques such as Bow, Motion Texture,Augmented-Bow and Sequential Motion Texture (SMT) by using leave-one-out cross-validation. The videos were based on the OSATS score(set by surgeons) and viewed by experts. [39]For the expert the OSATS score was set as 3, for intermediate it was 2, and for a beginner, it was set 1. The camera was mounted on the tripod, whereas the camera was mounted on the board vertical at 90-degree angle in our experiment. The cluster count taken in his experiment was 5, and for our experiment cluster count is 20. [39]

The results of his experiment were good when he used A-Bow as compared to BoW, but for Motion Texture, the accuracy was 83.3 percentage. In the end, he concluded that accuracy for an expert is lower than other two classes because the expert may not use the proper method of knot means not use all the steps to complete a knot and might develop a different style for the knot.[39] Here, the videos were analysed based on two classes, fft and bht with a different kernel and using different k clusters. For the cluster value more than 20, the interest points overlaps with each other and the output accuracy(near around 40%) was not good as compare to cluster k=20 or less than 20 (accuracy= 76% with linear kernel).

From the experiment, analysed that there are many reasons for the bad performance of the algorithm, like not enough data-sets(chose only those videos which were good),

light concentration on the videos, movement of the hands whether the surgeon is left-handed or right-handed, method of performing knots (which was analysed by the Irfan, as according to time and experience expert surgeon adapt different style to tie a knot while performing surgery).

Chapter 8

Future Work

This dissertation is based on the surgical videos divided into FFT and BHT to find out accuracy by using a classical approach of Machine Learning. However, this approach can be extended to divide data into 7 parts of a surgical knot and apply Multi-class SVM on it. Also, the number of videos or dataset can be increased in the next level because the data captured was very less for both egocentric and camera mounted on a stand. In the next iteration, a number of the clusters can be increased or decreased in order to find out the better results on it. In order to extend the project, expert surgeons data can be included to set the ground-truth value.

The study performed by Irfan Essa was also useful in terms of using OSATS score, but he can also extend the approach by developing data-driven gesture vocabularies of expert surgeons gesture. Irfan accepted that OSATS evaluation would give a positive effect on real-time world in medicinal schools and emergency clinics. [39]

Bibliography

[1] "www.uchealth.com/services/robotic-surgery/patient-information/davinci-surgical-system/,"

[2] "https://www.researchgate.net/figure/bag-of-words bow representation of the input that is next follow by softmax fig2 308884219,"

[3] "Moon v2v-posenet voxel-to-voxel prediction cvpr 2018 paper," 2018.

[4] "https://docs.opencv.org/3.4/dc/d0d/tutorial py features harris.html,"

[5] "http://www.micc.unifi.it/seidenari/wp-content/uploads/2010/01/a51-spatio-temporal-features1.pdf,"

[6] "https://en.wikipedia.org/wiki/k-means clustering,"

[7] "https://en.wikipedia.org/wiki/optical flow,"

[8] "https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb,"

[9] "https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496,"

[10] "https://www.analyticsvidhya.com/blog/2018/10/computer-vision-approach-hand-gesture-recognition,"

[11] T.-M. L. Hsi-Chieh Lee1, Che-Yu Shih2, "Computer-vision based hand gesture recognition and its application in iphone,"

[12] S. Awad, Liscum, "Does the subjective evaluation of medical student surgical knowledge correlate with written and oral exam performance?," *Journal of Surgical Research*, vol. 104, 2002.

[13] "https://www.animatedknots.com/surgical-tie-knot-one-hand-technique,"

[14] G. Martin, Regehr, "Objective structured assessment of technical skill (osats) for surgical residents.," *British Journal of Surgery*, vol. 84, 1997.

[15] "https://computer-vision-ai.com/blog/computer-vision-history,"

[16] J. L. R. Ankit Chaudhary, Sonia Raheja, "Intelligent approaches to interact with machines using hand gesture recognition in natural way: A survey," *IJCSES*, vol. 2, 2011.

[17] S. H. M. Soriano, B. Martinkauppi and M. Laaksonen, "Skin detection in video under changing illumination conditions," *Proceedings 15th International Conference on Pattern Recognition*, vol. 1, 2000.

[18] X. Li, "Gesture recognition based on fuzzy c-means clustering algorithm," department of computer science," *The University of Tennessee Knoxville*, 2003.

[19] O. Malima and Cetin, "A fast algorithm for vision-based hand gesture recognition for robot control," *IEEE 14th Signal Processing and Communications Applications, pp. 1–4*, 2006.

[20] M. M. V. Datta, S. Bann and A. Darzi, "The surgical efficiency score: a feasible, reliable, and valid method of skills assessment," *The American journal of surgery*, vol. 192, 2006.

[21] G. C. P. Dollar, V. Rabaud and S. Belongie., "Behavior recognition via sparse spatio-temporal features. in proceedings of the 14th international conference on computer communications and networks," *IEEE Computer Society*, 2005.

[22] T. T. Geert Willems and L. V. Gool., "An efficient dense and scale-invariant spatio-temporal interest point detector.," *n Computer Vision– ECCV*, 2008.

[23] V. R. Haro BB, Zappella L, "Surgical gesture classification from video data.," *International conference on medical image computing and computer-assisted intervention—MICCAI*, vol. 34–41, 2012.

[24] H. G. V. R. Zappella L, Béjar B, "Surgical gesture classification from video and kinematic data.," *Med Image Anal*, vol. 732–745, 2013.

[25] M. M. Twinanda, Shehata, "Endonet: A deep architecture for recognition tasks on laparoscopic videos.," *IEEE Trans Med Imaging*, vol. 36(1):86–97, 2017.

[26] L. H. G. DiPietro R, Vedula, "Recognizing surgical activities with recurrent neural networks.," *International conference on medical image computing and computer-assisted interventionl.*, vol. 551–558, 2013.

[27] U. K. L. S. Wang, H., "Evaluation of local spatio-temporal features for action recognition," *BMVC*, 2009.

[28] P. I. Bettadapura, Schindler, "Augmenting bag-of-words: Datadriven discovery of temporal and structural information for activity recognition.," *CVPR*, 2013.

[29] "https://en.wikipedia.org/wiki/deep learning,"

[30] "https://hal-mines-paristech.archives-ouvertes.fr/hal-01737771/file/deeplearning-handskeletalgesturerecognition-mines-paristech-fg2018.pdf,"

[31] Y. Wu and T. S. HuangI, "Hand modeling analysis and recognition for visionbased human computer interaction," *IEEE Signal Processing Magazine, Special Issue on Immersive Interactive Technology*, 2001.

[32] A. Corradini, "Real-time gesture recognition by means of hybrid recognizers," *Lecture Notes in Computer Science, Revised Papers from the International Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction*, 2001.

[33] R. Liang and M. Ouhyoung, "A real-time continuous gesture recognition system for sign language,," *Proc. 3rd International Conference on Automatic Face and Gesture Recognition*, 1998.

[34] Laptev, "https://www.irisa.fr/vista/papers/2003 iccv laptev.pdf,"

[35] "https://github.com/anirudhhkumarr/action-recognition/tree/master/stip-2.0-linux,"

[36] "https://blog.easysol.net/machine-learning-algorithms-3/,"

[37] "https://en.wikipedia.org/wiki/intel realsense,"

[38] "https://www.learnopencv.com/svm-using-scikit-learn-in-python/,"

[39] I. E. Andrew McCaskie, "Video based assessment of osats using sequential motion textures," *M2CAI*, 2014.

Appendix 1

```
#Path of the files
forefingerthrow_path = '/Users/khushboogoyal/Documents/python_files/egocenteric/bht/'
save = '/Users/khushboogoyal/Documents/python_files/egocenteric/bht/rgb/'

i = 0
for filename in os.listdir(forefingerthrow_path):

    if os.path.isfile(os.path.join(forefingerthrow_path, filename)):

        i+=1
        pipeline = prs.pipeline()
        # create a config object
        config = prs.config()

        prs.config.enable_device_from_file(config, os.path.join(forefingerthrow_path, filename))
        config.enable_stream(prs.stream.color, 640, 480, prs.format.rgb8, 15)
        #config.enable_stream(prs.stream.depth, 640, 480, prs.format.z16, 15)

        fht = os.path.join(save, str(i), '.avi')
        fftourcc = cv2.VideoWriter_fourcc(*'XVID')
        out = cv2.VideoWriter(fht,fftourcc, 20.0, (640,480))
        profile1 = pipeline.start(config)

        #depthsensor = profile1.get_device().first_depth_sensor()
        #depthscale = depthsensor.get_depth_scale()
```

```
26
27          #clippingdistance = 2 #1 meter
28          #clippingdistance = clippingdistance/ depthscale
29
30          #align_to = prs.stream.color
31          #align = prs.align(align_to)
32
33          try:
34              while True:
35                  f = pipeline.wait_for_frames()
36                  # Align the depth frame to color frame
37                  #aligned_frames = align.process(f)
38
39                  ## afterwards remove the alignement
40                  # Get aligned f
41                  #aligned_depth_frame = aligned_frames.get_depth_frame
    () # aligned_depth_frame is a 640x480 depth image
42                  color_frame = f.get_color_frame()
43
44                  # Validate that both f are valid
45                  #if not aligned_depth_frame or not color_frame:
46                  #continue
47
48
49                  cimage = np.asanyarray(color_frame.get_data())
50                  cimage = cv2.cvtColor(cimage, cv2.COLOR_BGR2RGB)
51
52                  # Remove background - Set pixels further than
    clipping_distance to grey
53                  #grey_color = 153
54                  #depth_image_3d = np.dstack((depth_image,depth_image,
    depth_image))
55                  #depth image is 1 channel, color is 3 channels
56                  # bg_removed = np.where((depth_image_3d >
    clipping_distance) | (depth_image_3d <= 0), grey_color, cimage)
57
58                  # Render images
59                  #dcolormap = cv2.applyColorMap(cv2.convertScaleAbs(
    depth_image, alpha=0.03), cv2.COLORMAP_JET)
```

```
60
61
62                  #images = np.hstack((bg_removed, dcolormap))
63                  #images = np.hstack((cimage, dcolormap))
64
65                  cv2.namedWindow('Align', cv2.WINDOW_AUTOSIZE)
66                  out.write(cimage)
67                  cv2.imshow('Align', cimage)
68
69                  key = cv2.waitKey(1)
70                  # Press esc or 'q' to close the image window
71                  if key & 0xFF == ord('q') or key == 27:
72                      cv2.destroyAllWindows()
73                      break
74
75                      # Check if new frame is ready
76                      #if pipeline.poll_for_frames(f):
77                      # cframe = f.wait_for_frames()
78                      #colorimage = np.asanyarray(cframe.get_data())
79                      #colorimage = cv2.cvtColor(colorimage, cv2.
    COLOR_BGR2RGB)
80                      #colorwriter.write(colorimage)
81
82          finally:
83              pipeline.stop()
```

Listing 1: Bag File Extraction

Appendix 2

```
import numpy as np
import sklearn
from sklearn.cluster import KMeans
import time
import pickle
import pandas as pd


#Selecting files from the folder
for i in range(1,56):
    y = np.genfromtxt("/Users/khushboogoyal/Downloads/try/f" +str(i)+".txt", delimiter=" ")
    np.savetxt("/Users/khushboogoyal/Downloads/f" +str(i)+".txt", y)

for i in range(1,56):
    y = np.genfromtxt("/Users/khushboogoyal/Downloads/try/b" +str(i)+".txt", delimiter=" ")
    np.savetxt("/Users/khushboogoyal/Documents/python_files/b" +str(i)+".txt", y)

with open("/Users/khushboogoyal/Downloads/b1.txt") as f:
        content = f.readlines()
floats=[]
for i in range(len(content)):
        f=[float(x) for x in content[i].split()]
        floats.append(f)
arr = np.array(floats)
arr = arr.astype(np.float16)
# Extracting hog-hof features from the csv file and save to another file
```

```
27
28 #dimensions = arr[:,1:9]
29 #np.savetxt("/Users/khushboogoyal/Documents/python_files/dimensions/
       fp15.csv",dimensions,delimiter=",")
30 hogHof = arr[:,9:]
31 np.savetxt("/Users/khushboogoyal/Downloads/hoghof/b1.csv",hogHof ,
       delimiter=",")
32
33 #Applying k-means on each csv file
34
35 for i in range(4,4):
36     f = pd.read_csv("/Users/khushboogoyal/Downloads/hoghof/b" +str(i)+
       ".csv")
37     f.shape
38     #f.head()
39     from sklearn.cluster import KMeans
40     import matplotlib.pyplot as plt
41     kmeans = KMeans(n_clusters=20).fit(f)
42     centroids = kmeans.cluster_centers_
43     #print(centroids)
44     import numpy as np
45     print(kmeans.labels_)
46     np.savetxt("/Users/khushboogoyal/Documents/python_files/kmean/fp"
       +str(i)+".csv",kmeans.labels_, delimiter=",")
47
48 #Read the file and check it by head function
49
50 k= pd.read_csv('/Users/khushboogoyal/Documents/python_files/kmean/fp1.
       csv')
51 k.head()
52
53 #Applying BoW on each file
54 # cluster values are 0-19
55 #bp=1 orginally it was fore fingre
56 #fp=0 orginally it was back
57
58 import array as arr
59 for j in range(1,3):
60     with open("/Users/khushboogoyal/Documents/python_files/kmean/fp"+
```

```
    str(j)"+.csv") as csvFile:
61          content = csvFile.read()
62          x=np.array([])
63          myCsvRow="0"
64          for i in range(20):
65              y= content.count(str(i))
66              x=np.append(x,y)
67              print(y)
68              myCsvRow = myCsvRow +","+str(y)
69          print(myCsvRow)
70      #print(x)
71      with open("/Users/khushboogoyal/Documents/python_files/occur/fp"+
    str(j)"+.csv","w") as fd:
72          fd.write(myCsvRow)
73          fd.write("\n")
74 #np.savetxt("/Users/khushboogoyal/Documents/python_files/occ.csv",x,
    delimiter=',')
75
76 #Merge all files into one, to pass it in SVM
77
78 import csv
79 #reader = csv.reader(open("/Users/khushboogoyal/Documents/python_files
    /occur/fp1.csv"))
80 reader = csv.reader(open("/Users/khushboogoyal/Documents/python_files/
    merge/f44.csv"))
81 reader2 = csv.reader(open("/Users/khushboogoyal/Documents/python_files
    /occur/fp45.csv"))
82 f = open("/Users/khushboogoyal/Documents/python_files/merge/f45.csv",
    "w")
83 writer = csv.writer(f)
84 for row in reader:
85     writer.writerow(row)
86 for row in reader2:
87     writer.writerow(row)
88 f.close()
89
90 # Applying SVM with different kernels
91
92 import pandas as pd
```

```
93 import numpy as np
94 np.random.seed(48)
95 f= pd.read_csv('/Users/khushboogoyal/Documents/python_files/merge/
       merge.csv')
96 X = f.drop('label', axis=1)
97 y = f['label']
98
99 from sklearn.model_selection import train_test_split
100 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
       0.20)
101
102 #Plot heatmap for the f.corr()
103 import seaborn as sns
104 import matplotlib.pyplot as plt
105 plt.figure(figsize=(20,12))
106 sns.heatmap(f.corr(), annot=True)
107
108 #kernel = liner
109 from sklearn.svm import SVC
110 from sklearn.metrics import classification_report, confusion_matrix,
       accuracy_score, average_precision_score
111 svclassifier = SVC(kernel='poly', random_state=20, gamma='auto')
112 svclassifier.fit(X_train, y_train)
113 y_pred = svclassifier.predict(X_test)
114 average_precision_score= average_precision_score(y_test,y_pred)
115 print('Average precision-recall score: {0:0.2f}'.format(
       average_precision_score))
116 #print(confusion_matrix(y_test,y_pred))
117 accuracy_score= accuracy_score(y_test,y_pred)
118 print('Average accuracy score: {0:0.2f}'.format(accuracy_score))
119
120 #kernel = poly
121 from sklearn.svm import SVC
122 from sklearn.metrics import classification_report, confusion_matrix,
       accuracy_score, average_precision_score
123 svclassifier = SVC(kernel='poly', random_state=20, gamma='auto')
124 svclassifier.fit(X_train, y_train)
125 y_pred = svclassifier.predict(X_test)
126 average_precision_score= average_precision_score(y_test,y_pred)
```

```
127 print('Average precision-recall score: {0:0.2f}'.format(
        average_precision_score))
128
129 #kernel = sigmod
130 #print(confusion_matrix(y_test,y_pred))
131 accuracy_score= accuracy_score(y_test,y_pred)
132 print('Average accuracy score: {0:0.2f}'.format(accuracy_score))
133 svclassifier = SVC(kernel='sigmoid', random_state=20, gamma='auto')
134 svclassifier.fit(X_train, y_train)
135 y_pred = svclassifier.predict(X_test)
136 average_precision_score= average_precision_score(y_test,y_pred)
137 print('Average precision-recall score: {0:0.2f}'.format(
        average_precision_score))
138 #print(confusion_matrix(y_test,y_pred))
139 accuracy_score= accuracy_score(y_test,y_pred)
140 print('Average accuracy score: {0:0.2f}'.format(accuracy_score))
141
142 #ROC curve
143 from sklearn.metrics import precision_recall_curve
144 import matplotlib.pyplot as plt
145 from inspect import signature
146 precision, recall, _ = precision_recall_curve(y_test,y_pred)
147 step_kwargs = ({'step': 'post'}
148                if 'step' in signature(plt.fill_between).parameters
149                else {})
150 plt.step(recall, precision, color='b', alpha=0.2,where='post')
151 plt.fill_between(recall, precision, alpha=0.2, color='b', **
        step_kwargs)
152 plt.xlabel('Recall')
153 plt.ylabel('Precision')
154 plt.ylim([0.0, 1.05])
155 plt.xlim([0.0, 1.0])
156 plt.title('2-class Precision-Recall curve: AP={0:0.2f}'.format(
        average_precision_score))
157
158 #Plot confusion Matrix
159
160 confusion= confusion_matrix(y_test,y_pred)
161 print(confusion)
```

```
sns.heatmap(confusion, annot=True)

#k-fold crossvalidation where k=3

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix,
   accuracy_score, average_precision_score
from sklearn import svm
param_grid = {'C': [0.1, 1, 10, 100], 'gamma': [1, 0.1, 0.01, 0.001,
   0.00001, 10]}
 # Make grid search classifier
clf_grid = GridSearchCV(svm.SVC(), param_grid, verbose=1)
 # Train the classifier
clf_grid.fit(X_train, y_train)
confusion= confusion_matrix(y_test,y_pred)
print(confusion)
sns.heatmap(confusion, annot=True)
# clf = grid.best_estimator_()
print("Best Parameters:\n", clf_grid.best_params_)
print("Best Estimators:\n", clf_grid.best_estimator_)
y_pred = svclassifier.predict(X_test)
average_precision_score= average_precision_score(y_test,y_pred)
print('Average precision-recall score: {0:0.2f}'.format(
   average_precision_score))
#print(confusion_matrix(y_test,y_pred))
accuracy_score= accuracy_score(y_test,y_pred)
print('Average accuracy score: {0:0.2f}'.format(accuracy_score))
#0.00001
```

Listing 2: Code for k-means and BoW and SVM