Content Based Video Retrieval using Local Ternary Pattern Feature

by

SHAH SHEEL RIKESHKUMAR

(201911011)

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY

in

INFORMATION AND COMMUNICATION TECHNOLOGY

to

Dhirubhai Ambani Institute of Information and Communication Technology

December,2022

Declaration

I hereby declare that

i the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii due acknowledgment has been made in the text to all the reference material used.

11 /01/2023

Shah Sheel Rikeshkumar

December,2022

Certificate

This is to certify that the thesis work entitled "Content Based Video Retrieval using Local Ternary Pattern Feature" has been carried out by Shah Sheel Rikeshkumar (201911011) for the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology under my supervision.

MKhare 11/01/23

Prof. Manish Khare
Thesis Supervisor

Acknowledgements

On the successful completion of my M.Tech thesis work, I am really grateful to all the individuals who have assisted me during this beautiful journey and without whom this work would not have been possible.

Firstly, I would like to express my sincere gratitude towards my thesis supervisor, Prof. Manish Khare for their constant support and guidance throughout the journey. I would like to convey my sincere thanks to Prof. Manish Khare for giving me a chance to explore this field and providing valuable assistance in carrying out this work. His friendly and helping nature and excellent domain knowledge enabled me to achieve the best outcome. His technical and domain knowledge genuinely helped me in solving any issue I had during this journey. The discussions made through continuous progress meetings truly motivated me to explore new things to achieve the desired outcome.

I would like to extend my gratitude to the Help Desk, DAIICT, for providing access to the required resources to carry out my work and giving all the technical assistance and help whenever needed. I would also like to thank Resource Center, DAIICT, for providing credentials to all the reading materials for the literature survey of this work.

Furthermore, I thank all my friends for their support and inspiration throughout this journey. A special thanks to my family for their unconditional love, care, encouragement, and confidence in me.

Contents

Abstract

In the 21st century, image and video repositories have been increasing drastically. This is attracting visual based smart city solutions for transport, healthcare, healthcare, safety, hospitality, sports visuals, etc. Effective storage of Video database and its retrieval is the new problem to solve. This search analyzes metadata like keywords, captions, titles, etc. With current multimedia solutions of computer vision and advanced image processing, the CBVR approach uses content understanding of images like color, shape descriptor, textures, deep features, etc. Limitations in inheritance of metadata systems have led to content understanding based approaches. This work proposes to generate feature vectors for each of the data base and query videos. We've proposed to use a color feature based PCC distance for video shot detection and key frame extraction to remove redundancy or dimensionality reduction. Further uses local ternary pattern (LTP) and uniform local ternary pattern dynamic texture feature on key frames and feature vector generation. Then Euclidean distance with KNN classifier for video retrieval. We've utilized UCF50 human action data set for the proposed work.

Keywords: Content based video retrieval, Video shot detection, Key Frame extraction, Local ternary pattern, Local binary pattern, Feature vector generation

List of Figures

List of Tables

Abbreviations

CBVR	**C**ontent **B**ased **V**ideo **R**etrieval
CBIR	**C**ontent **B**ased **I**mage **R**etrieval
PCC	**P**earson's **C**orrelation **C**oefficeint
KF	**K**ey **F**rame
ZNCC	**Z**ero mean **N**ormalized **C**orrelation **C**oefficeint
FV	**F**eature **V**ector
LOA	**L**ion's **O**ptimization **A**lgorithm
FG	**F**oreground
RGB	**R**ed **G**reen **B**lue
HSV	**H**ue **S**aturation **V**alue
LBP	**L**ocal **B**inary **P**attern
LBP-TOP	**L**ocal **B**inary **P**attern based on **T**hree **O**rthogonal **P**lanes
LTP	**L**ocal **T**ernary **P**attern
KNN	**K** **N**earset **N**ighbour
SD	**S**tandard **D**eviation
YOLO	**Y**ou **O**nly **L**ook **O**nce

Symbols

M	Mean of the image
σ	Standard deviation of the image
σ^2	Variance of the image
fv	feature vector
$P_{i,y}$	Probability or intensity of y$^{th}colorchannelofi^{th}pixel$
α_y	mean of y$^{th}colorchannel$
β_y	SD of y$^{th}colorchannel$
SK_y	Skewness of y$^{th}colorchannel$
K_y	Kurtosis of y$^{th}colorchannel$
$\mu_{m,n}$	shape moment of (m,n)$^{th}order$
$\psi_{m,n}$	(m,n)$^{th}ordermomentweighingkernel$
$\eta_{m,n}$	scale invariant shape moment of (m,n)$^{th}order$
h_i	Hu moment of i$^{th}order$
fv	feature vector
Z_n^m	Zernike Moment of (m,n)$^{th}order$
R_n^m	Zernike Polynomial of (m,n)$^{th}order$
g_p	Pixel intensity of p$^{th}neighborpixel$
g_c	Pixel intensity of center pixel

Chapter 1

Introduction

Content based video retrieval has been used increasingly to describe the process of retrieving videos from large data-bases with its content understanding. Content understanding, feature vector generation, feature matching and video retrieval are key points of our proposed work alongside dimensionality reduction of frame feature vector. There were many advanced image processing based and deep learning based approaches for CBVR. Better content understanding with decreased computation is the key problem to solve here.

1.1 Motivation

Effective visual content understanding has been done in many type of frame or image features like color moments, shape moments, texture features, SIFT, SURF, edge or key point descriptors, etc alongside deep learning based features that utilises CNN type networks. Different features have their own methods for computation. Though present some drawbacks and may take higher computation time. With many type of features and its FV generation with effective retrieval algorithm is an important and interesting problem. Videos are matched and retrieved by similarity, dissimilarity or distance of its feature vectors. That leads to machine learning approach for faster and better retrieval.

1.2 Objective

- Identifying superior features that can handle vast type of video situations is our primary goal.
- Reducing the feature vector dimension while retaining a majority of the information and designing a robust retrieval approach is our main objective.
- Earlier approaches have some drawbacks such as : deep learning approaches always require re-training with an update of the video data set, some features can not handle varying lighting conditions, moving objects in the video, different scenes, etc.
- Our objective is to design an approach that can handle different scenes, varying lighting conditions, multiple objects and can work in different environment.

1.3 Contributions

The key contributions of this work are as follow :

- We propose use the PCC based approach for video shot detection that segments each video into multiple small video clip.
- We propose to use color moment based key frame extraction that extracts a single key frame that has the most visual information from each video shot.

- We propose to use LTP feature for each key frame that will lead to FV of dimension (2,256) for each frame.
- We propose to use KNN for video retrieval, Euclidean distance metric is used for KNN.
- We analyze this problem in method similar to video classification for further work and improvisation.

1.4 Thesis Organization

The remainder of the thesis is organized as follows :

- **Chapter 2** shows the literature surveys of previous research on CBVR
- **Chapter 3** explains the proposed approach
- **Chapter 4** shows the experimental results along with a comparison with existing approaches
- **Chapter 5** Finally we've concluded the work and have provided a further scope of improvement

Chapter 2

Literature Survey

(Spolaor et al. 2020) [3] suggests that most of the approaches uses video segmentation followed by key frame extraction for dimensionality reduction. Further color, shape, texture, etc. moments are extracted as features. Machine learning models are used for video retrieval.

2.1 Video Shot detection and Key frame extraction

Extracting frame features for all the video frames would become much more computationally heavy and with high data redundancy. Thus, most visual content frames are extracted as key frame or in some approaches candidate key frames. For this, videos are segmented into different video shot. There are 2 types for video shots gradual shot and abrupt shot. Gradual and abrupt shot are shown in Figure 2.1 and Figure 2.2.

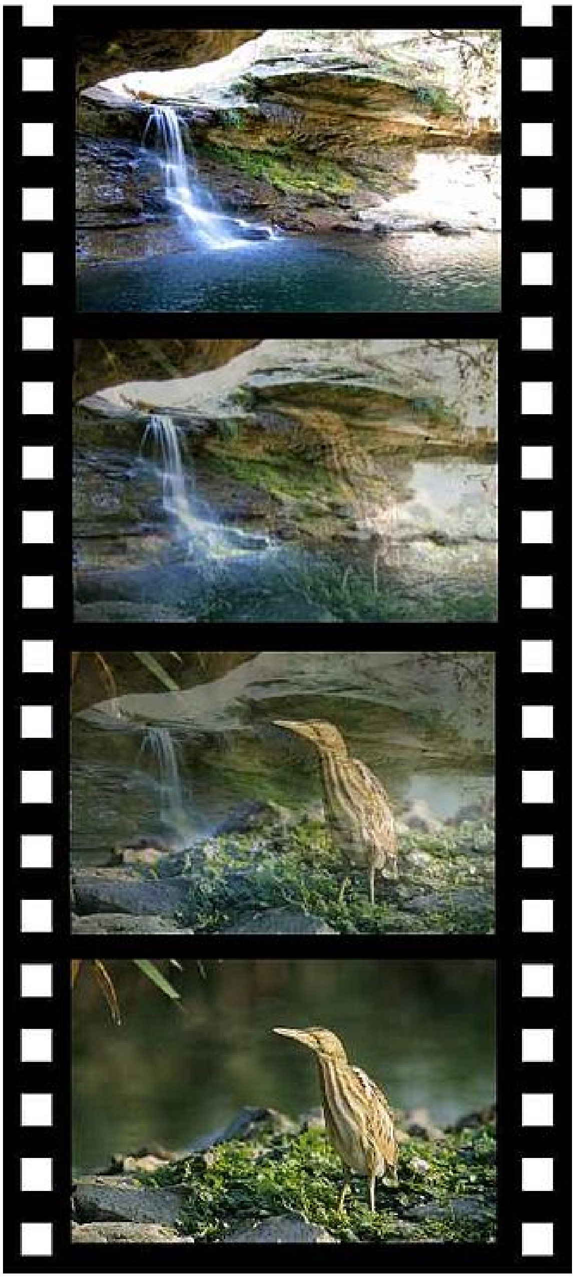


Figure 2.1: Gradual shot transition

(Spolaor et al. 2020) [3] suggests that cut based video shot detection is the most used approach. This approach works frame by frame and computes distance or similarities or dissimilarities of a pair of consecutive frames (frame-i and frame-i+1). This term is considered as scoring and with help of fixed threshold values or dynamic threshold

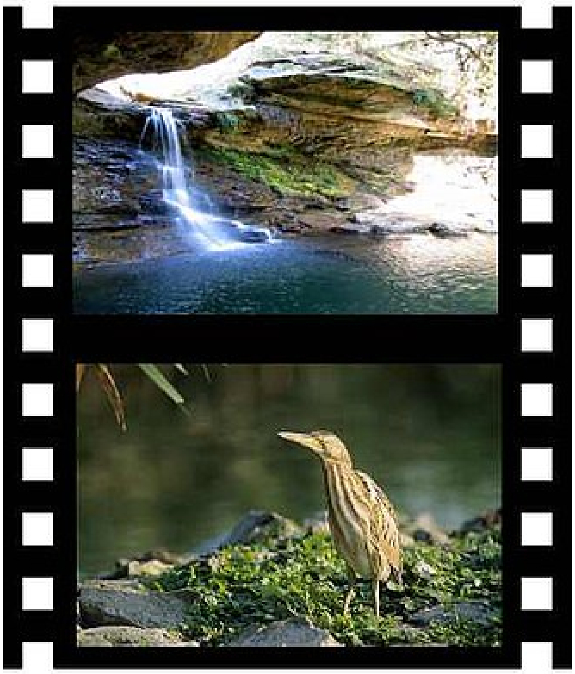


FIGURE 2.2: Abrupt shot transition

values based on some statistical algorithm or machine learning based model, video shot transitions are detected. This term is referred as decision.

Further for each key frames of a video, a set of image features are computed and ultimately this generated the feature vector for a video. Then videos are retrieved based on the distance based or similarity based method using appropriate ML model [3].

2.1.1 Significant Frame Detection

(Zhao et al. 2021) [1] has proposed to use a significant frame detection approach that is similar to **LOA** : Lion's Optimization Algorithm. It based on the living style of Lions which is social by nature. This approach is based on color moments Mean and SD. For each frames of the video shot, background is removed and only foreground is considered as interesting part Figure (2.3). Then distance between consecutive frames of the video shot are computed(2.1). Then fitness value is computed with (2.2) where $\alpha = 2$ is constant.

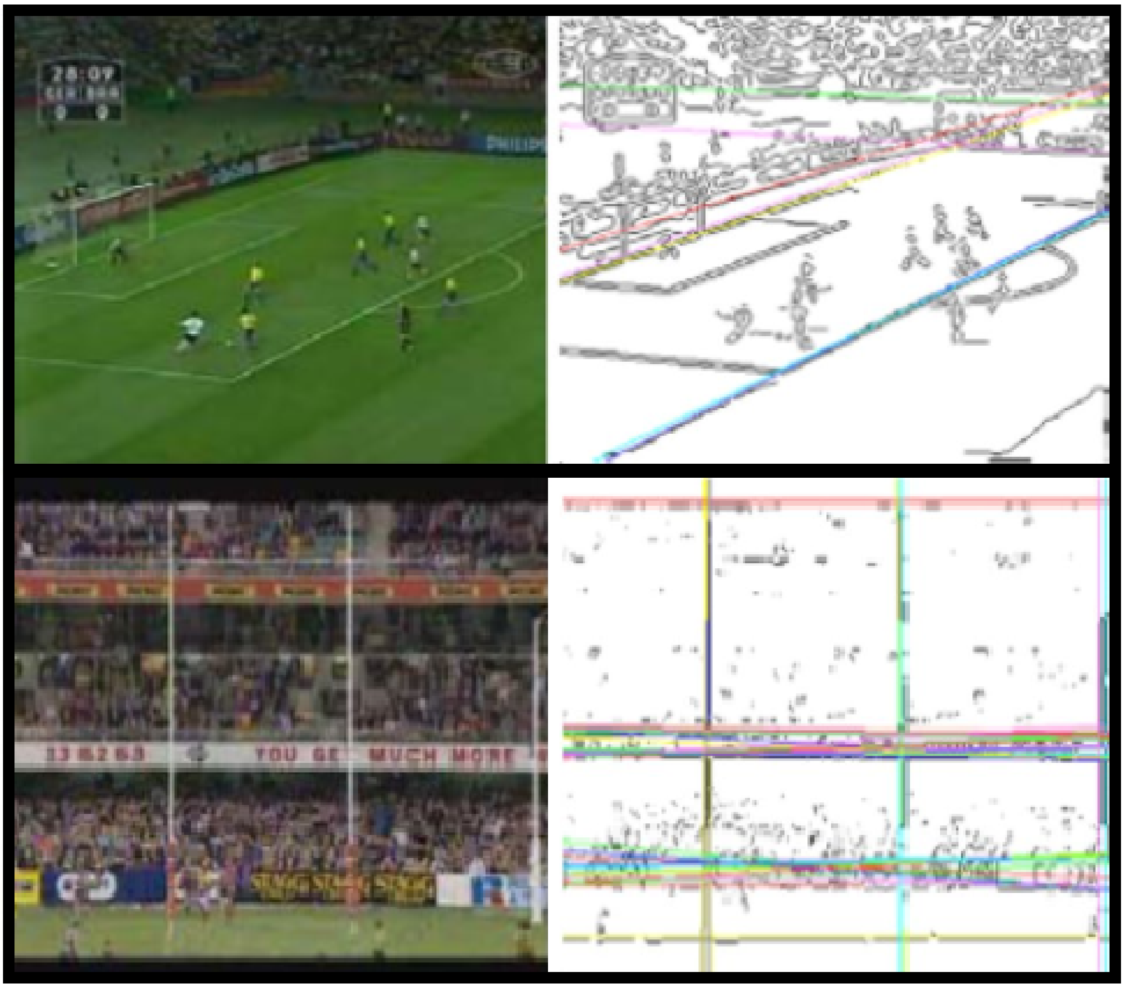


FIGURE 2.3: Background Elimination

$$D(p) = \sum_{r} \sum_{c} (FG_p - FG_{p+1}) \tag{2.1}$$

$$fv = M + \alpha \times SD \tag{2.2}$$

Here, $D(p)$ is the distance between foregrounds of frame-p and frame-p+1, FG_p is foreground image of frame-p, fv is fitness value, M and SD are mean and standard deviation of the $D(p)$.

Higher the fitness value, higher is the visual change between consecutive frames. So frame with highest fitness value is considered as significant frame from the video shot [1].

2.2 Image Features

There are plenty of features for videos or images like low and high level features. All these features are of different dimensions. In Figure (2.4) show various features and their dimensions [4] . Edges, blob, corners, etc. are low level features. Shape, object, etc. are high level features. Low level features are extracted through advanced image processing techniques while high level features are extracted using machine learning or deep learning techniques.

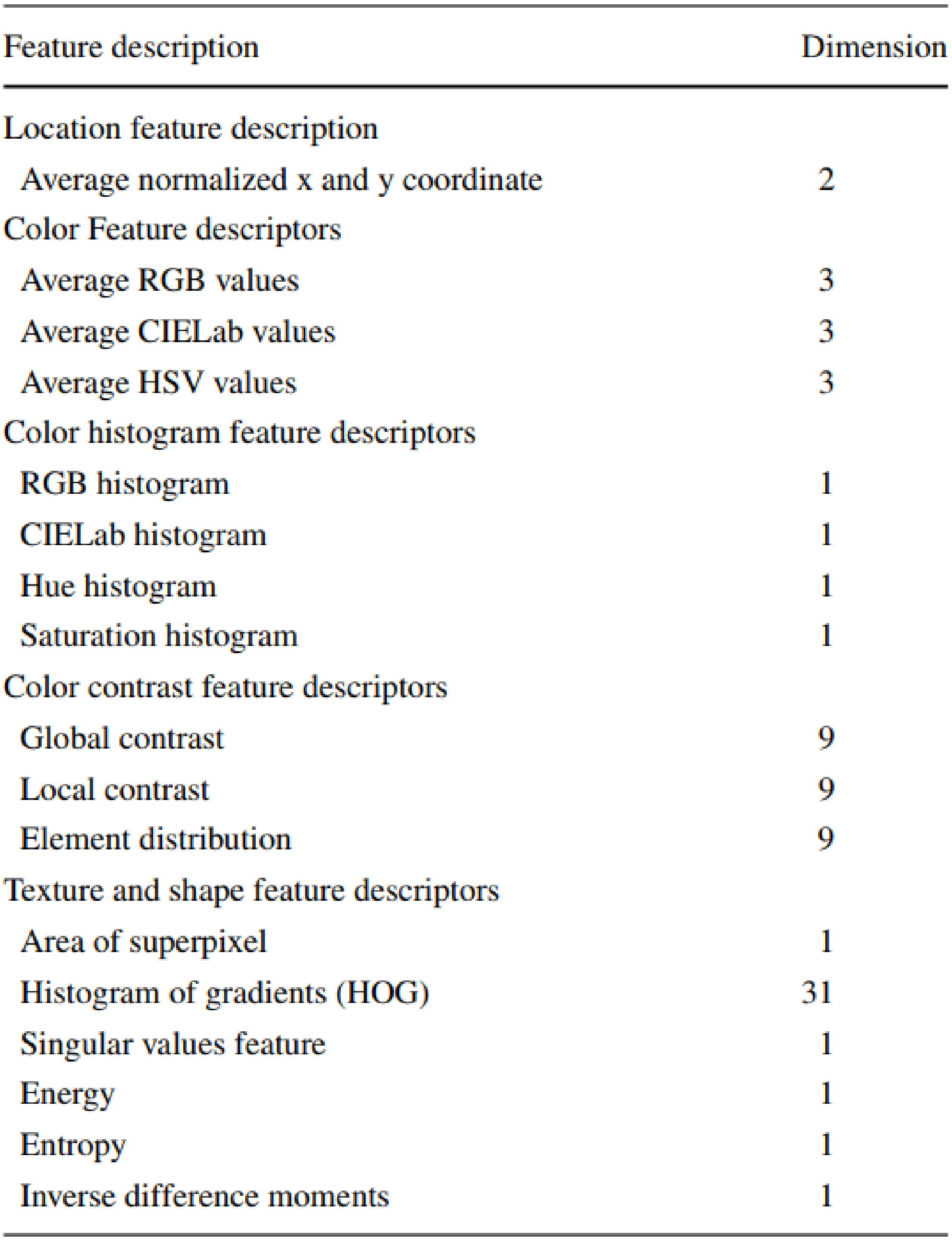
2.2.1 Color Features

Color features are generally referred as color moments like mean (2.3), standard deviation (2.4), skewness (2.5) and kurtosis (2.6) can be extracted with advanced image processing with pixel by pixel method.

These features collectively describe the color and its distribution in the image.

$$\alpha_y = \frac{1}{N}\sum_{i=1}^{N} P_{i,y} \tag{2.3}$$

$$\beta_y = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_{i,y} - \alpha_y)^2} \tag{2.4}$$



Feature description	Dimension
Location feature description	
Average normalized x and y coordinate	2
Color Feature descriptors	
Average RGB values	3
Average CIELab values	3
Average HSV values	3
Color histogram feature descriptors	
RGB histogram	1
CIELab histogram	1
Hue histogram	1
Saturation histogram	1
Color contrast feature descriptors	
Global contrast	9
Local contrast	9
Element distribution	9
Texture and shape feature descriptors	
Area of superpixel	1
Histogram of gradients (HOG)	31
Singular values feature	1
Energy	1
Entropy	1
Inverse difference moments	1

Figure 2.4: Generalised Image Features

$$SK_y = \sqrt[3]{\frac{1}{N}\sum_{i=1}^{N}(P_{i,y} - \alpha_y)^3} \tag{2.5}$$

$$K_y = N * \frac{\sum_{i=1}^{N}(P_{i,y} - \alpha_y)^4}{(\sum_{i=1}^{N}(P_{i,y} - \alpha_y)^2)^2} \tag{2.6}$$

α_y is mean of the image for color channel y. $P_{i,y}$ is the pixel intensity of i^{th} pixel for y^{th} color channel. y is color channel R,G,B or H,S,V or any other that we're focusing to utilise. All these color moments signifies different type of color information of the image. Like mean indicates average color of the image or say how dark or bright the image is if in gray scale or towards which color image is intended to. Standard deviation shows the how much the color is dispersed in relation to mean. Skewness indicates how much symmetrical the color distribution is. Kurtosis is computed as ratio of fourth color moment to variance (SD squared). Skewness and kurtosis helps to understand where the most of the information is lying in the image.

2.2.2 Shape Features

Shape features describes the shape of different objects like people, car, toy, table, utensils, etc. with help of number or set of numbers and referred as shape moments. Equation (2.7) shows the the shape moment equation, where $\psi_{m,n}$ is the mn^{th} order moment weighing kernel or basis set, $I(x,y)$ is the pixel intensity of pixel (x,y) and $\mu_{m,n}$ is the mn^{th} order shape moment. In general, shape moments are weighted average of the image pixel pixel intensities.

$$\mu_{m,n} = \sum_{x=0}^{X} \sum_{y=0}^{Y} \psi_{m,n} * I(x,y) \tag{2.7}$$

Shape moments are computed for objects in the image. So object extraction or object is the key task here. For these there are multiple types of deep learning based architectures like RCNN, Fast-RCNN, Faster-RCNN, SSD, YOLO, etc. The latest and most widely used architecture recently is YOLOv3 (You only look once). It is used in real-time object detection. YOLOv3 is CNN based architecture consisting of 106 layers. 53 CNN layers and 53 pooling layers. It detects bounding boxes objects in 3 different deep neural CNN levels. It gives outcome of 1-D array of length (5+c) for each bounding box object. c here stands for number of classes on which YOLO is trained. In rest 5, 4 will stand for center pixel coordinate, height, width and objectness P_0 score. Objectness score shows the total probability of having or detecting an object in the bounding box. YOLOv3 deep architecture is in Figure (2.5)

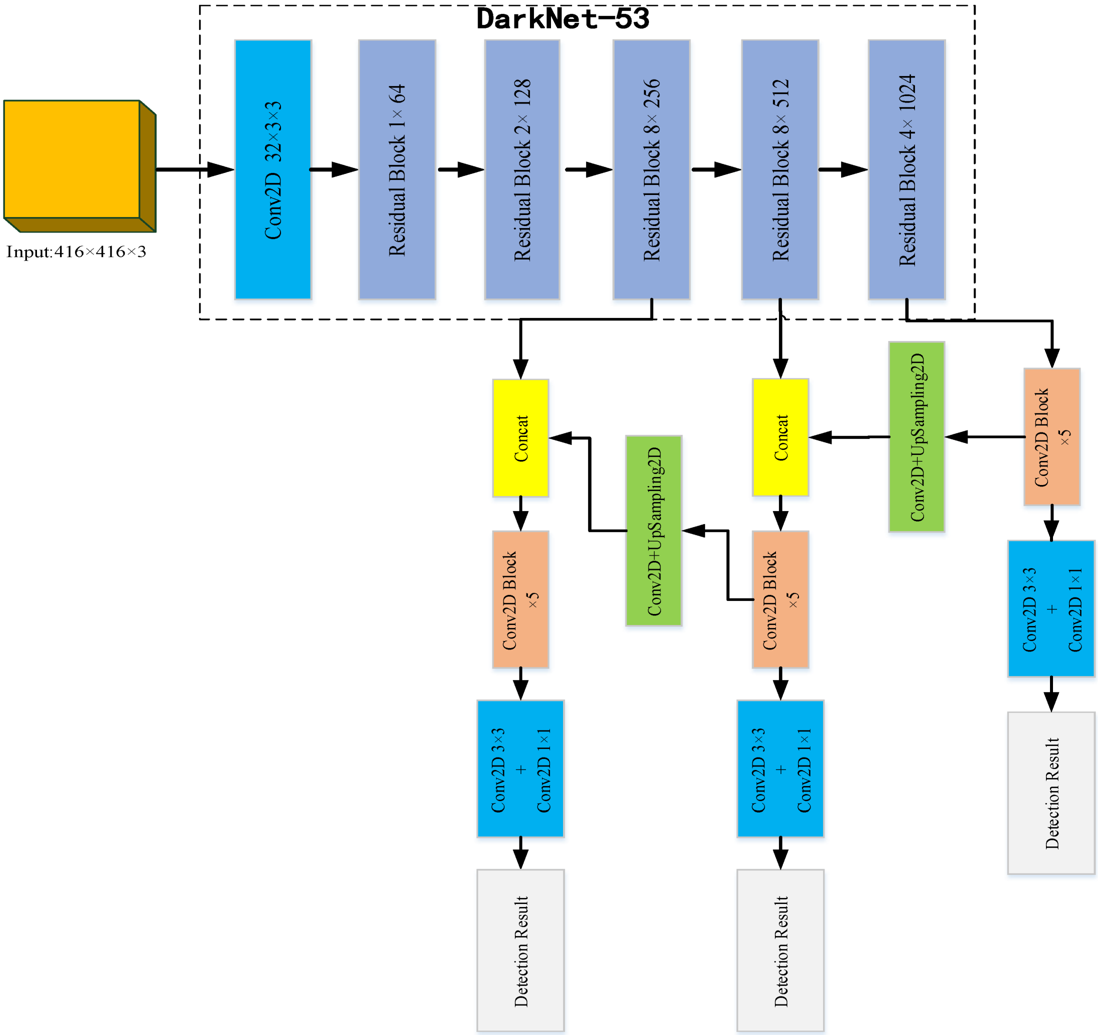


FIGURE 2.5: YOLO v3 Architecture

Figure (2.6) shows the object detection image for the frame from **BenchPressv01c01** video. As we can see it has detected 5 objects. All object class are **Person** For this object detection we've utilised **YOLOv3** deep neural architecture model.

- Probability Threshold = 0.5 (Detects an object if its probability in the bounding box is greater than or equal to 0.5
- nms Threshold = 0.3 (Removes the overlapping bounding boxes if intersection area is greater than or equal to 0.3), (lesser the value, lesser the number of bounding boxes detected, more stricter to overlapping regions)

We've utilised pre-trained YOLOv3-416 [5]. The 416 is the size of target image to process in the model. Image of size 416 × 416.

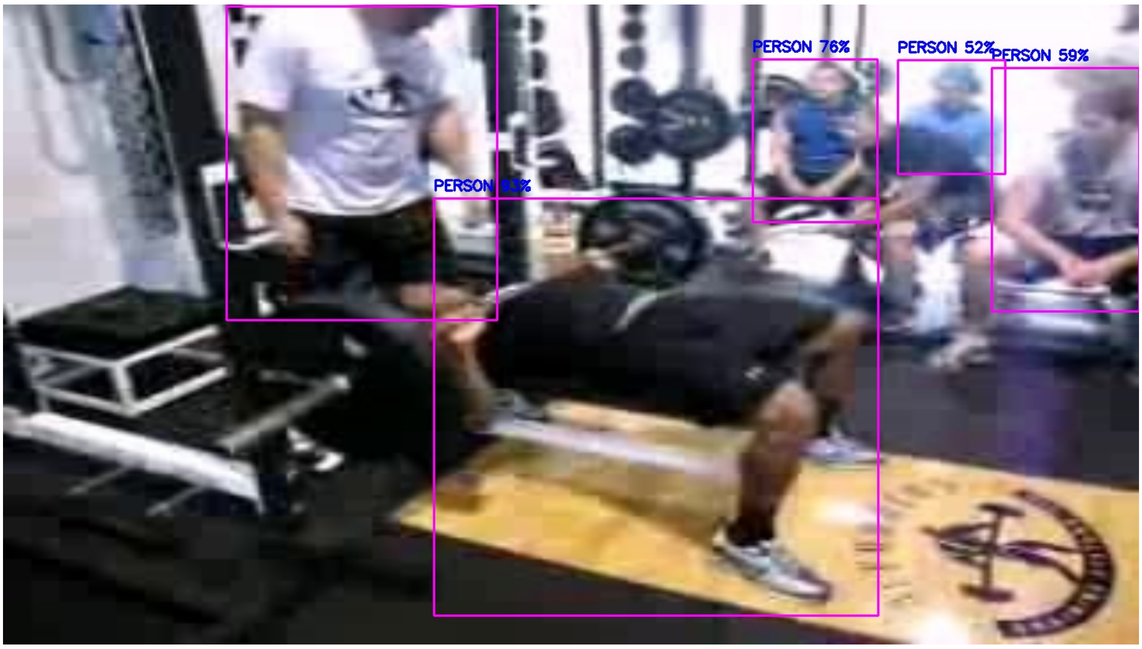


Figure 2.6: Object Detection for a frame of BenchPressv01c01 video

These image moments might vary with the change in size and orientation. So matching set of moments for same type of objects would be much more difficult with these moments. So invariant shape moments has been introduced. Where $\psi_{m,n} = x^n y^m$ are the basis set. These moments (2.8) are rotation, scale and translation invariant. It solely depends on shape of the object, not on its size or orientation.

$$\mu_{m,n} = \sum_{x=0}^{X}\sum_{y=0}^{Y} x^n y^m * I(x,y) \tag{2.8}$$

These moments will still depend on the position of the shape. So we often use central moments. (2.9) These are transformation invariant and position invariant image shape moment. Here, $\mu_{m,n}$ is the mn^{th} order shape moment.

$$\mu_{m,n} = \sum_{x=0}^{X}\sum_{y=0}^{Y} (x-\bar{x})^n (y-\bar{y})^m * I(x,y) \tag{2.9}$$

Further scale invariant moments are obtained with equation (2.10). Where $\mu_{0,0}$ stands for mean of the image (2.3). $\eta_{m,n}$ is the mn^{th} order central shape moment.

$$\eta_{m,n} = \frac{\mu_{m,n}}{\mu_{0,0}^{\frac{m+n}{2}+1}} \tag{2.10}$$

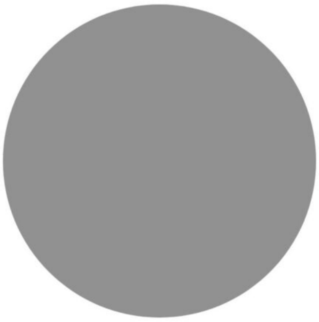


Figure 2.7: Circle Shape Object for Shape Moment

Hu moments are set of 7 central moments that are invariant of image transformation [6]. First 6 Hu moments are translation, scale and shape invariant. 7^{th} Hu moment changes sign with object reflection(2.11).

$$
\begin{aligned}
h_0 &= \eta_{2,0} + \eta_{0,2} \\
h_1 &= (\eta_{2,0} - \eta_{0,2})^2 + 4\eta_{1,1}^2 \\
h_2 &= (\eta_{3,0} - 3\eta_{1,2})^2 + (3\eta_{2,1} - \eta_{0,3})^2 \\
h_3 &= (\eta_{3,0} + \eta_{1,2})^2 + (\eta_{2,1} + \eta_{0,3})^2 \\
h_4 &= (\eta_{3,0} - 3\eta_{1,2})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2] \\
&\quad + (3\eta_{2,1} - \eta_{0,3})(\eta_{2,1} + \eta_{0,3})[3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] \\
h_5 &= (\eta_{2,0} + \eta_{0,2})[(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2] \\
&\quad + 4\eta_{1,1}(\eta_{3,0} + \eta_{1,2})(\eta_{2,1} + \eta_{0,3}) \\
h_6 &= (3\eta_{2,1} - \eta_{0,3})(\eta_{3,0} + \eta_{1,2})[(\eta_{3,0} + \eta_{1,2})^2 - 3(\eta_{2,1} + \eta_{0,3})^2] \\
&\quad - (\eta_{3,0} - 3\eta_{1,2})(\eta_{2,1} + \eta_{0,3})[3(\eta_{3,0} + \eta_{1,2})^2 - (\eta_{2,1} + \eta_{0,3})^2]
\end{aligned}
\tag{2.11}
$$

For Figure 2.7 Hu moments are as follows Table (2.1) :

TABLE 2.1: Hu Moments for Circle image Figure 2.7

μ_0	2.71233687e-01
μ_1	4.35405029e-13
μ_2	2.14390260e-13
μ_3	1.72007108e-13
μ_4	3.08589210e-26
μ_5	-4.83257349e-20
μ_6	-1.17803196e-26

These shape moments, invariant moments, centralised moments, Hu moments all have some extent of data redundancy as its basis $x^n y^m$ are not orthogonal to each other. So Zernike moments are used as shape moments as they are orthogonal and invariant to rotation and translation [1]. There are even and odd Zernike moments(2.12).

$$\begin{aligned} Even : Z_n^m(\rho,\phi) &= R_n^m(\rho)cos(m\phi) \\ Odd : Z_n^{-m}(\rho,\phi) &= R_n^m(\rho)sin(m\phi) \end{aligned} \tag{2.12}$$

$R_n^m(\rho)$ is Zernike polynomials(2.13). These are basis set of Zernike moments.

$$R_n^m(\rho) = \sum_{k=0}^{\frac{n-m}{2}} \frac{(-1)^k(n-k)!}{k!(\frac{n+m}{2}-k)!(\frac{n-m}{2}-k)!}\rho^{n-2k} \tag{2.13}$$

ρ is the radial distance of the object pixel that ranges in [0,1]. ϕ is the Azimuthal angle (angular measurement in the spherical coordinate system). Zernike polynomials are sequence of polynomials that are orthogonal on the unit disk. Zernike moment ranges in [0,1]. Lower order Zernike moments describes the general or overall outline shape of the object. Higher order Zernike moments cover more detailed aspect of the object shape [7].

For Figure 2.7 first 25 Zernike moments are as follows Table 2.2:

Table 2.2: Zernike Moments for Circle image Figure 2.7

3.1830e-01	4.2027e-04	8.3234e-07	4.1617e-07	8.4054e-04
3.6631e-10	7.9577e-01	2.0808e-06	3.9788e-01	1.8912e-03
1.5760e-03	1.5760e-03	2.9131e-06	2.3425e-06	1.1140e+00
3.6414e-06	2.5216e-03	3.7824e-03	3.7824e-03	6.4105e-09
1.07427e+00	1.5656e-05	7.1618e-01	6.5546e-06	1.7904e-01

As we can observe unlike Hu moments, Zernike moments are in range [0,1].

For Figure 2.6, as there are 5 objects detected, Zernike moment feature vector would be of size (5,25). As lower order Zernike moments describe overall shape of the object, we'll utilise Zernike moments in 2 ways : All 25 values and other feature vector using first 10 values.

2.2.3 Texture Feature

Texture features are used to quantify the perceived texture of an image [1]. Has proposed to use GIST descriptor to compute texture features. There are some dynamic textures also [2].

For GIST descriptor, Gabor filters are utilised. It is a convolution filter representing a combination of Gaussian and a sinusoidal term(2.14).

$$g(x, y, \lambda, \theta, \psi, \sigma, \gamma) = exp(-\frac{x'^2 + \gamma^2 y'^2}{2})exp(i(2\pi\frac{x'}{\lambda} + \psi)) \tag{2.14}$$

$$x, y : position\ in\ kernel$$

$$\lambda : wavelength$$

$$\theta : orientation\ of\ the\ filter$$

$$\psi : phase\ offset$$

$$\sigma : SD\ of\ gaussian\ envelope \approx 0.1$$

$$x' : xcos\theta + ysin\theta$$

$$y' : -xsin\theta + ycos\theta$$

GIST descriptors are compute in 4 scales and 8 orientation of angle hence total of 32 values for an image.

Dynamic texture is generally defined as texture with motion. These are mostly used in video still works on frame level. Local binary pattern (LBP) is the type of dynamic textures. LBP works with pixel by pixel fundamental. It computes a gray level texture 8-bit binary pattern for the grey-scale image. As the LBP feature assigns neighbor pixels into 2 levels, this is referred as binary pattern.

$$
\begin{aligned}
LBP_{p,r} &= \sum_{p=0}^{p-1} S(g_p - g_c) * 2^p \\
S &= \begin{cases} 0 & g_p < g_c \\ 1 & g_p \geq g_c \end{cases}
\end{aligned}
\tag{2.15}
$$

The LBP feature doesn't change with illumination changes and is rotation invariant. It can even remains nearly unchanged with moving object in a video. For each pixel, a binary pattern is computed with help of (2.15). Each neighbors are compared with the central pixel and assigned a bit, that circularly creates an 8 bit binary pattern. For Figure 2.8, LBP image is computed as Figure 2.9. LBP feature produces an grey scale image and further 256 bin histogram Figure 2.10 is computed and saved as LBP feature.

Figure 2.8: Lena Image

As this LBP feature is 256 bin histogram, feature vector size would be a bit higher. So further only uniform binary patterns are considered [2]. Circular binary patterns which have maximum 2 bit changes around a cycle are considered as uniform binary pattern. That leads to 58 bins only (56 patterns with 2 bit changes and 2 patterns with 0 bit changes). Indices of uniform patterns are as follows : [0, 1, 2, 3, 4, 6, 7, 8, 12, 14, 15, 16, 24, 28, 30, 31, 32, 48, 56, 60, 62, 63, 64, 96, 112, 120, 124, 126,

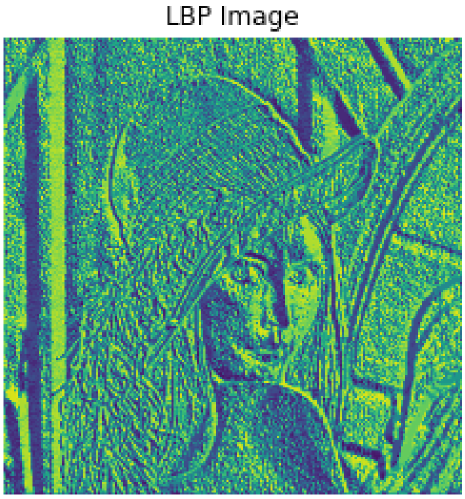


Figure 2.9: LBP Image for lena Image Figure 2.8

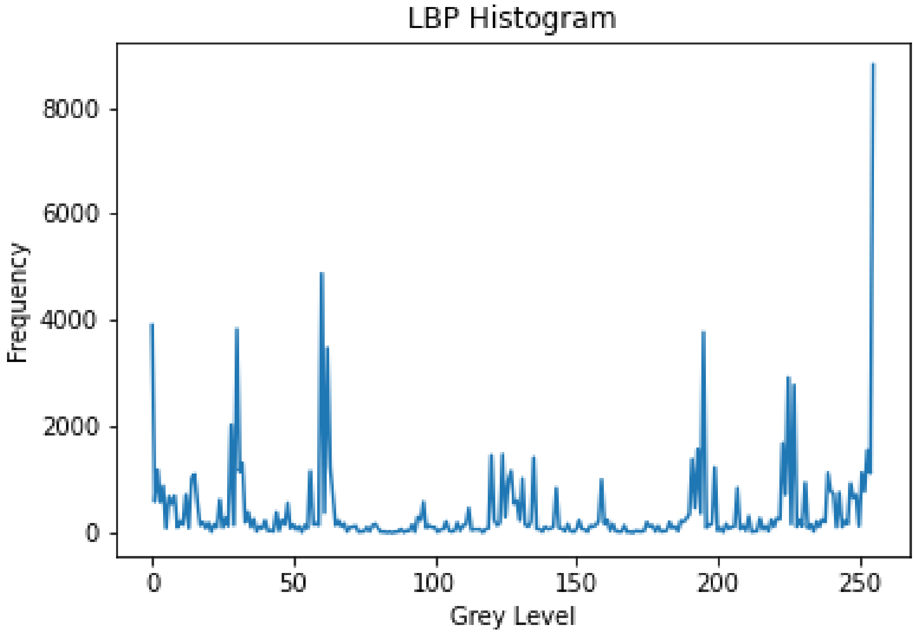


Figure 2.10: LBP Histogram for lena Image Figure 2.8

127, 128, 129, 131, 135, 143, 159, 191, 192, 193, 195, 199, 207, 223, 224, 225, 227, 231, 239, 240, 241, 243, 247, 248, 249, 251, 252, 253, 254,255]. Rest of the bins are summed in another bin, converting it to 59 bin histogram. In most of the cases this 59^{th} bin is also omitted [2]. This 58 bin histogram for Lena image Figure 2.8 is plotted in Figure 2.11.

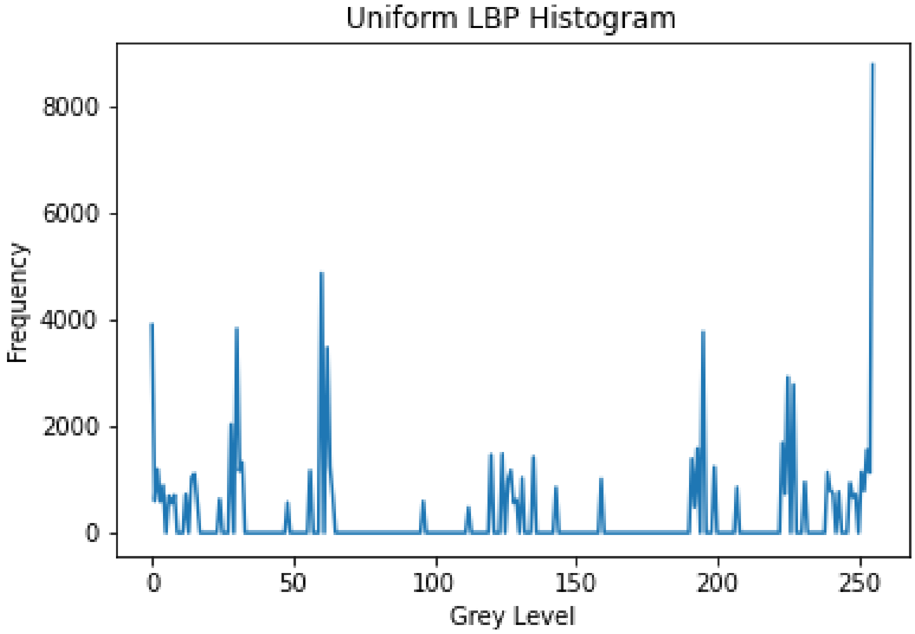


FIGURE 2.11: Uniform LBP Histogram for lena Image Figure 2.8

2.3 Feature Matching

We've utilised existing approach using color feature, shape feature and texture feature. For shape feature matching, we've used cross distance method. As we are using Zernike moments (all moments range in [0,1]) as shape moments, we can use Euclidean distance method for distance computation between 2 objects.

$$Dis(obj_1, obj_2) = \sqrt{\sum_{i=1}^{z}(Arr_1[i] - Arr_2[i])^2} \tag{2.16}$$

With Equation (2.16), we can compute distance between shape moments of 2 objects. Here z is number of Zernike moments we want to utilise and Arr_1,Arr_2 are shape feature vector for two objects. As different frames have different number of objects detected.

Let say $Frame_i$ has detected 5 objects and $Frame_j$ has 3 objects detected. We'll compute $Dis(obj_1, obj_2)$ (2.16) between 1st object from $Frame_j$ to each object of $Frame_i$ and will further proceed with the least distance. Like wise is done for all 3 objects of $Frame_j$. Here this type of cross distance (2.17) is used because object detected are in different order for all the frame. So we need to match similar type of objects. Here, M is the minimum number of detected objects among $Frame_i$ and $Frame_j$. Assumed $Frame_i$ has lower number of object, unless swapped. P is the maximum number of detected objects among $Frame_i$ and $Frame_j$.

$$dis(Frame_i, Frame_j) = \frac{1}{M}\sqrt{\sum_{m=1}^{M} min(Dis(obj_{i,m}, obj_{j,p=[0,P]}))^2} \tag{2.17}$$

Chapter 3

Proposed Method

We propose an approach for CBVR, that utilises the local ternary pattern (LTP) as image feature. As shown in the block diagram in Figure 3.1 system generates a feature vector (FV) for each video. Feature vectors are computed for data base videos and stored. Then for a query video, feature vector is computed and videos are retrieved from data base with help of FV matching with distance method or other machine learning approach.

3.1 Video Shot Detection

We've used a color based frames similarity measurement Pearson's Correlation Coefficient (PCC) [8] for shot boundary detection Figure 3.2. It computes PCC for each pair of consecutive frames in a video(3.1).

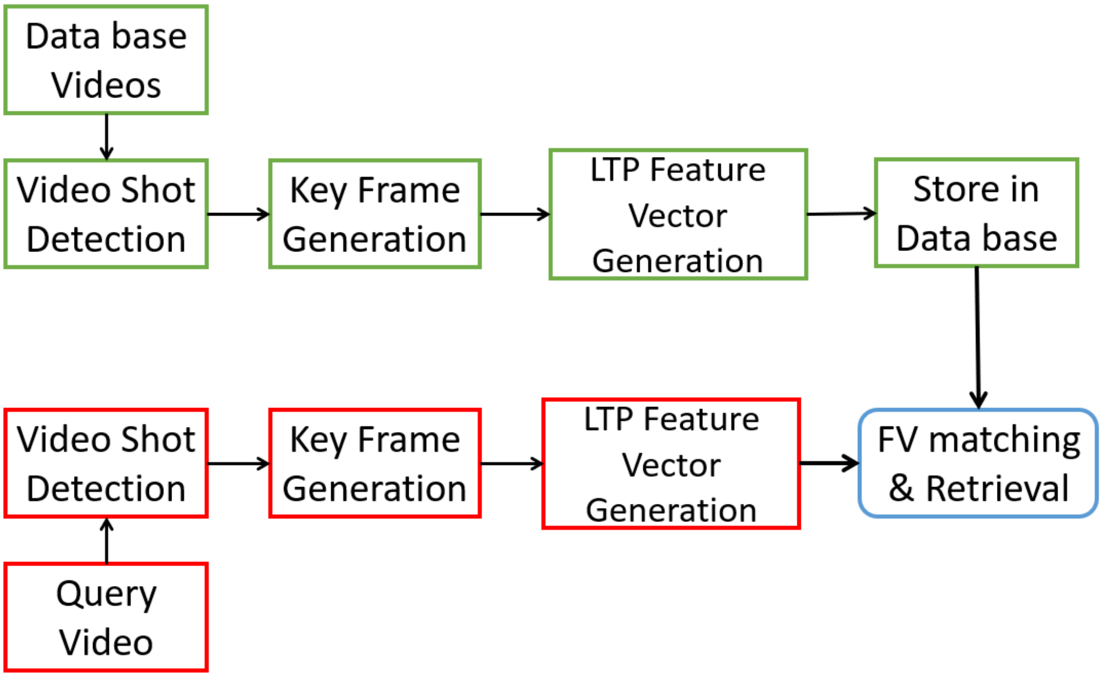


Figure 3.1: Proposed Approach

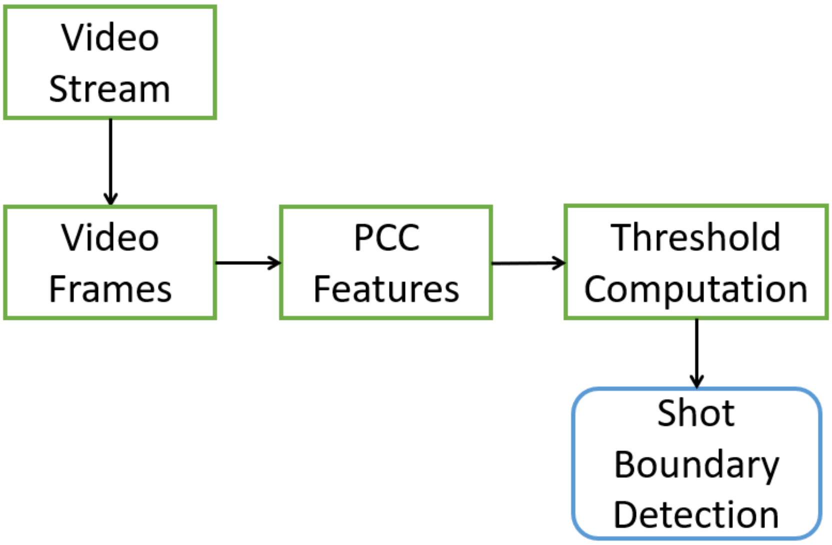


Figure 3.2: Shot Boundary Detection Block Diagram

$$PCC = \frac{\sum_{i=1}^{MN}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{MN}(x_i - \bar{x})^2 \sum_{i=1}^{MN}(y_i - \bar{y})^2}} \tag{3.1}$$

PCC value ranges between $[-1,1]$. Value 1 indicates highest similarity or exact correlation. Value 0 shows zero correlation between two images and value -1 shows negative correlation.

There are two type of shot detection, abrupt and gradual. For abrupt change, there would be a sudden drop in the PCC value. So it can be detected easily. For gradual change, there would not be a sudden drop in PCC value so we've to utilise an algorithm that can perform this task.

For this, PCC value is computed for each of 3 R,G,B color channels for each pair of consecutive frames and stored in F_R,F_G,F_B. Then its mean and variance are computed as μ_R, μ_G, μ_B, σ_R, σ_G, σ_B. Then threshold values are computed for all 3 color channels as shown in equation (3.2).

$$\begin{aligned} T_{P_R} &= \mu_R + (\alpha * \sigma_R) \\ T_{P_R} &= \mu_G + (\alpha * \sigma_G) \\ T_{P_B} &= \mu_B + (\alpha * \sigma_B) \end{aligned} \tag{3.2}$$

The frames for which PCC values are less than threshold values are considered as shot boundary. The values of α ranges between [0,1]. For value it'll give more refined shot boundaries resulting into more number of shots. That makes sure that no information left behind in this part. Value of α is empirically taken as 1 [8].

3.2 Key Frame Extraction

There are so many frames in a video. Utilising all the frames' feature would be quite a process costly task. So video shot detection and key frame extraction algorithms are implemented. Out key frame extraction approach extracts a single key frame from each video shot detected.

For the video frame which have most dominant pair of (m, σ) is considered as key frame. Pareto-dominance algorithm is used to find the dominant frame. Frame F1 is dominant than frame F2, if (m_1, σ_1) is greater than (m_2, σ_2) or any 1 of (m_1, σ_1) is greater than F2, another parameter not lesser than F2.

The frame having highest Pareto dominance count is considered as key frame.

Figure 3.4 shows the key frames extracted from trial video from PizzaTossing class. PCC based shot detection algorithm has produced so many shots from a small videos, hence more key frames resulting into large feature vector. So we've extracted a Super Key frame (SKF) considering whole video as a single shot. For this we've skipped the video shot detection part Figure 3.3.

FIGURE 3.3: Super Key Frame for example video

3.3 Feature Extraction

Previously LBP-TOP (Local binary pattern) based dynamic texture feature has been used for the image or video retrieval problem. We've proposed an approach

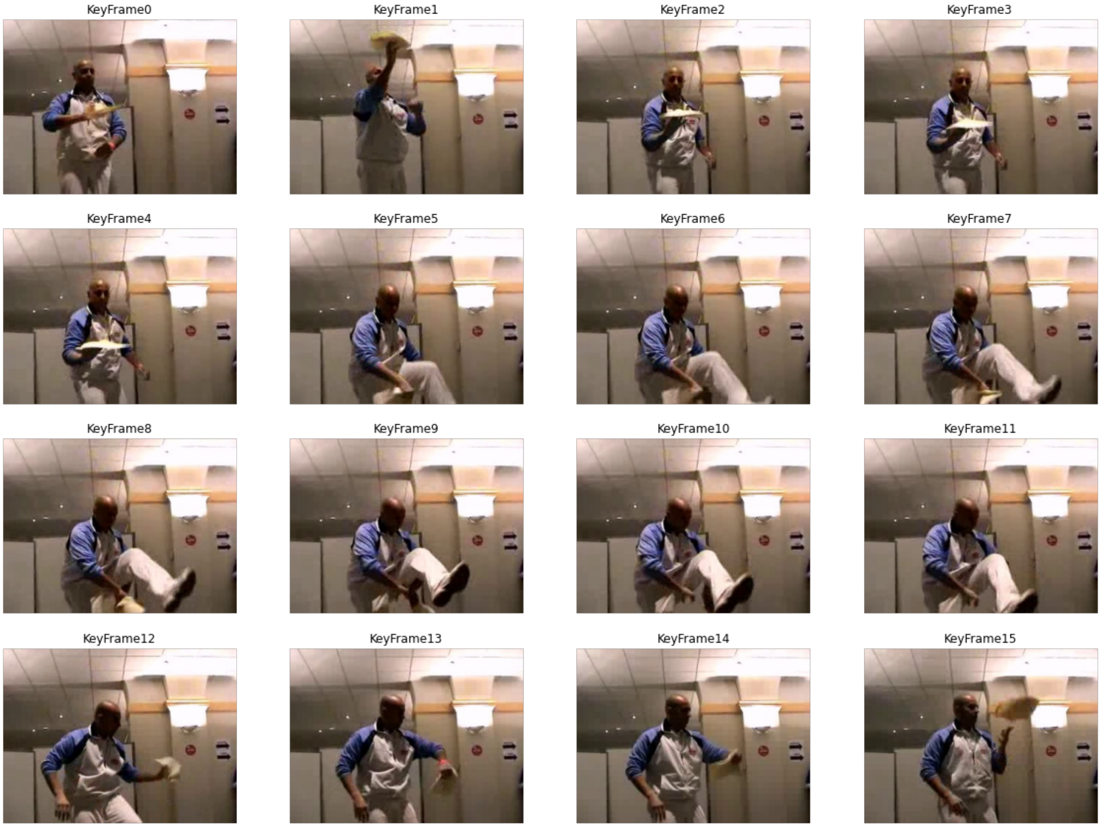


FIGURE 3.4: Key frames from Pizza Tossing example video

here by using Local ternary pattern (LTP) dynamic texture. LTP is less sensitive to noise than LBP as it threshold with some intensity apart from centre pixel. Still retains rotation invariant, moving object detection properties[9].

Unlike LBP, LTP thresholds each neighboring pixels into 3 level, [-1,0,1] 3.3. Computing a histogram for ternary pattern would result into large range. So 2 binary patterns are computed LTP_U, LTP_L [10]. Figure 3.3 shows the LTP computation. Figure 3.6 shows the LTP images for Figure 2.8. Histogram of these 2 LTP binary patterns are computed and further utilised as feature vector as shown in Figure 3.7 and Figure 3.8. As we can see 0 bin frequency is much higher than rest, that might dominate the distance computation, matching and retrieval. So we've plotted histogram skipping 0 bin as shown in Figure 3.9 and Figure 3.10.

$$LTP_{p,r} = \sum_{p=0}^{p-1} S(g_p - g_c) * 2^p \tag{3.3}$$

$$S = \begin{cases} -1 & g_p - g_c < -k \\ 0 & |g_p - g_c| < k \\ 1 & g_p - g_c > k \end{cases} \tag{3.4}$$

- g_c is the center pixel gray level intensity
- g_p is the neighbor pixel gray level intensity
- k is the threshold defined by the user, k=5 in below Figure 3.5

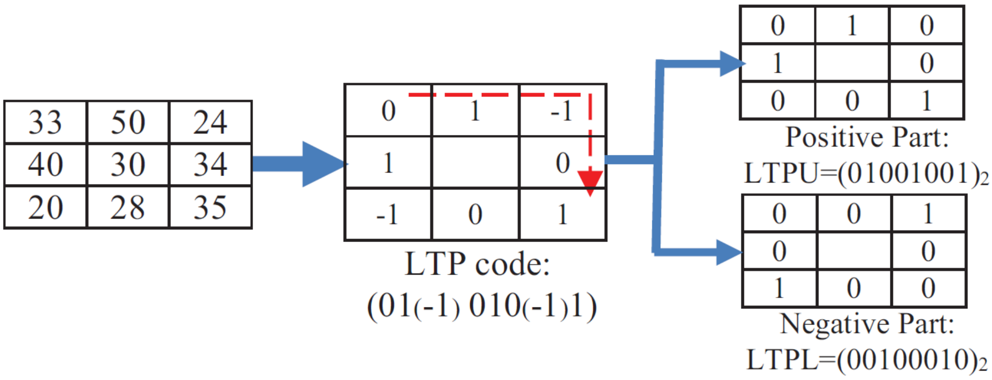


FIGURE 3.5: LTP Pattern

These 2 LTP pattern histograms are stored as feature vector(FV).

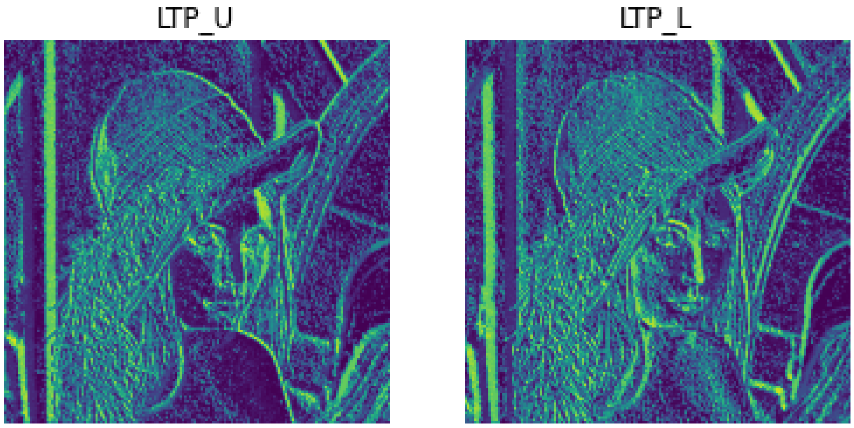


FIGURE 3.6: LTP Image for Lena image Figure 2.8

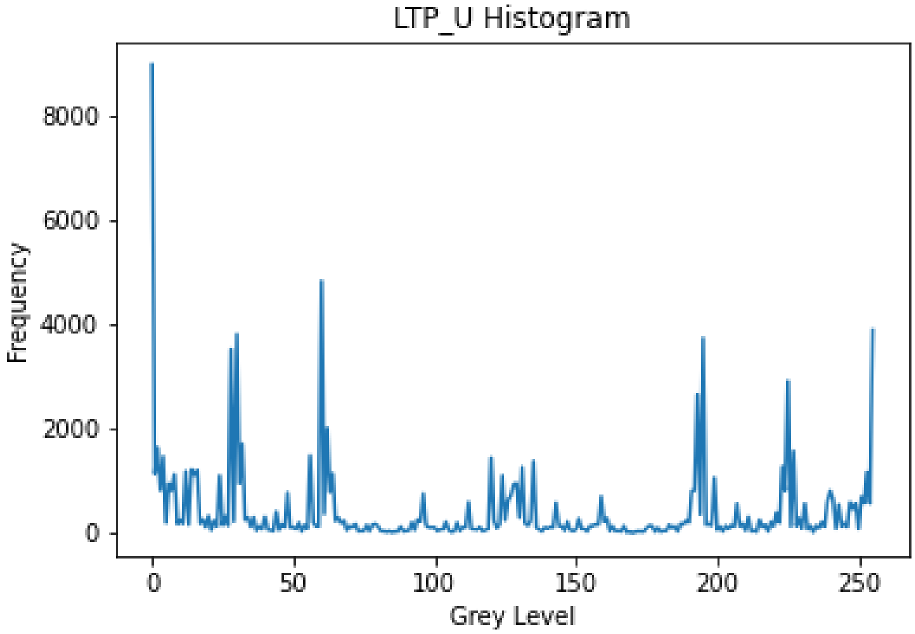


FIGURE 3.7: LTP-U Histogram with 0-255 bins for Lena image Figure 2.8

3.4 Matching and Retrieval

Now for video retrieval, distance between query video FV and data base video FV are computed for each of data base videos. In the equation given below, distance between K key frames are computed. K is the minimum number of key frame among all the videos. As there are 2 LTP patterns and 256 bin histogram, histogram distance is

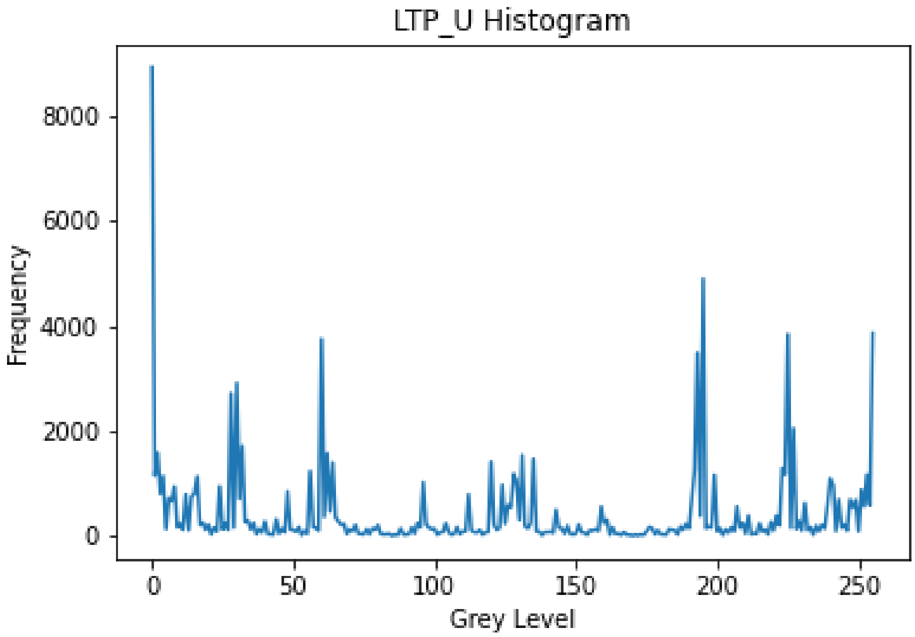


FIGURE 3.8: LTP-l Histogram with 0-255 bins for Lena image Figure 2.8

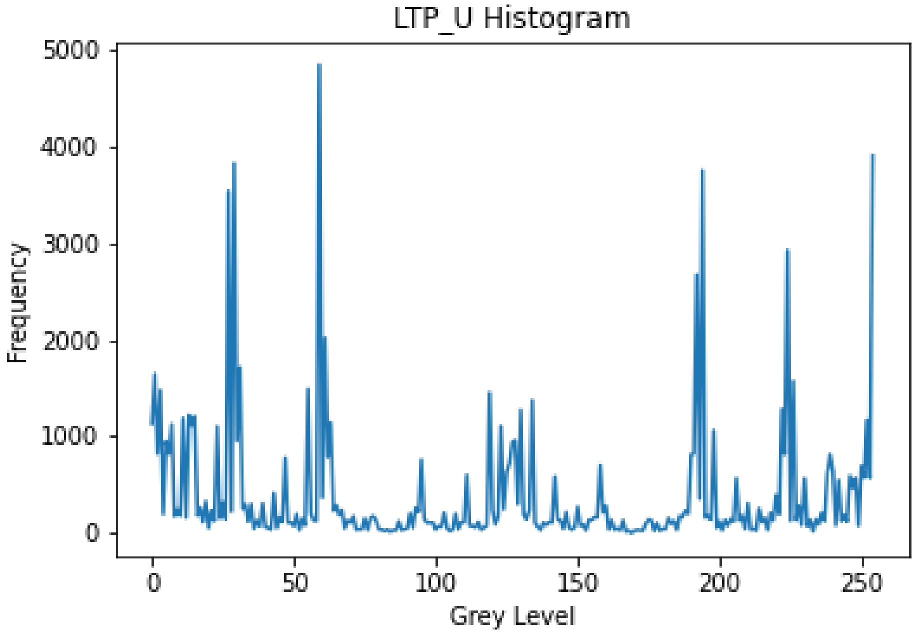


FIGURE 3.9: LTP-U Histogram with 1-255 bins for Lena image Figure 2.8

computed as follows:

$$D_{loc} = \sqrt{\sum_{l=1}^{K}\sum_{j=0}^{1}\sum_{l=0}^{255}(FV_{Q_{l,j}}(i) - FV_{D_{l,j}}(i))^2} \tag{3.5}$$

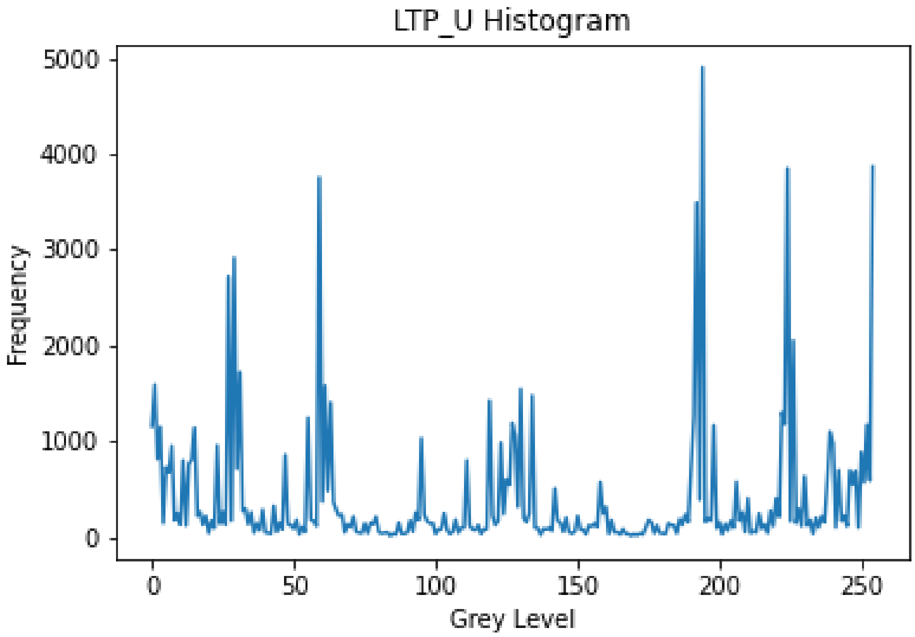


FIGURE 3.10: LTP-l Histogram with 1-255 bins for Lena image Figure 2.8

Here, FV_Q and FV_D are feature vector for query and database videos, $j = 0, 1$ for LTP_U and LTP_L histograms. Lower the distance higher the rank is for the retrieval. We've used K nearest neighbor classifier for the retrieval with K=5. For this we've considered retrieval problem as similar to the video classification problem. This problem is evaluated with classification accuracy measure. For conventional retrieval problem, we can utilise rank accuracy for performance measurement.

3.4.1 Retrieval Videos for a random video from data set

Here we've retrieved 10 videos for a video **BenchPressg01co1.avi** from **BenchPress** class. Retrieved videos are in Table (3.1).

Table 3.1: Retrieved Video for example video

Rank	Video Name
1	vBenchPressg01c02.avi
2	vBenchPressg03c07.avi
3	vPizzaTossingg03c02.avi
4	vBenchPressg04c01.avi
5	vBenchPressg03c04.avi
6	vPizzaTossingg05c01.avi
7	vBenchPressg04c03.avi
8	vBenchPressg03c06.avi
9	vBenchPressg03c05.avi
10	vPizzaTossingg01c01.avi

As we can see, total of 7 videos are retrieved from the same class **BenchPress** out of 10 videos. Rest of 3 videos are retrieved from **PizzaTossing** class.

Chapter 4

Results and Discussion

4.1 Data Set

We've performed and compared our proposed approach on the UCF50 Human action video data set [11]. It consists of 50 human action classes, each having 100-200 videos. All the videos are of duration ranging from2s to 25s.

We've utilised total of 7 classes named in (4.1). All are randomly selected. Further for balanced data set we've randomly selected 40 videos from each of the human action chances.

Table 4.1: Human Action Class

BenchPress
Diving
GolfSwing
HorseRiding
JumpRope
PizzaTossing
ThrowDiscus

4.2 Results

Rotation invariant binary patterns are called as uniform binary patterns. For evaluation we've matched whole histogram array and uniform binary pattern's histogram array. In LTP feature, as -1 and 1 are floored to 0 in either of LTP_U or LTP_L, 0 bin frequency is very dominant over other that we can see in the histogram plots. So we've matched [1,255] bin histogram skipping 0 bin frequency. We've performed this problem as similar to video classification problem.

- Plan 1 : Color, Shape feature (4.2) (Pratibha et al. 2020) [1]
- Plan 2: LBP-TOP with all KFs Table (4.3) (Reddy et al. 2022) [2]
- Plan 3: LTP with all KFs, K=5 Table (4.4)
- Plan 4: LTP with all KFs, K=10 Table (4.5)

Table 4.2: Plan 1 : Color Shape Moment, (Pratibha et al. 2020) [1]

No. of Zernike Moments	Accuracy
Z=25	63.57%
Z=10	66.78%
Z=5	56.42%

Table 4.3: Plan 2 : LBP-TOP, (Reddy et al. 2022) [2]

Bin	Accuracy
[0,2550]	60.357%
Uniform	59.64%

Table 4.4: Plan 3 : LTP (K=5 for KNN), Proposed Approach

K Width	Bins	Accuracy
1	[0,255]	61.43%
1	[1,255]	66.43%
1	Uniform	67.5%
2	[0,255]	57.14%
2	[1,255]	72.85%
2	Uniform	71.07%
3	[0,255]	54.64%
3	[1,255]	70.12%
3	Uniform	68.57%

Table 4.5: Plan 4 : LTP (K=10 for KNN), Proposed Approach

K Width	Bins	Accuracy
1	[0,255]	61.43%
1	[1,255]	57.267%
1	Uniform	56.87%
2	[0,255]	62.739%
2	[1,255]	67.53%
2	Uniform	60.428%
3	[0,255]	64.45%
3	[1,255]	62.57%
3	Uniform	54.307%

For Plan 1, Table (4.2) shows as we lower the number of Zernike's moments to be used for feature matching, accuracy increases. As higher order Zernike's moments are used to describe more details of the detected objects, it will affect distance much with minor change in the object of same class. But with very low number of Zernike's moments will give worse results. As it will match with just outer shape of the objects without any details being matched.

As we can observe from Table (4.4) and Table (4.5), considering video retrieval problem as similar to classification problem with KNN, K=5 has shown better results than KNN, K=10. So we can conclude that the more videos we retrieve the result deteriorates. Higher the rank of retrieved video, higher the chance of retrieving lesser similar video.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

As pre-trained YOLOv3 object detection has only 80 object classes. [5] It may miss out so many detailed objects that may define video class more specifically. So custom trained model with higher number of object classes with detailed information should be used. Advanced object detection model might solve this issue to some extent.

LTP feature with $K = 2$ and histogram bins $[1, 255]$ skipping 0 bin frequency has given the best result amongst all the approaches.

LTP feature based approach is collectively more accurate than LBP-TOP approach at all. Increasing K value in LTP has shown poor performance as it nullify some of the objects having low dynamic range in the frames.

Utilising only uniform binary patterns has shown a little lower performance than $[1, 255]$ bin histogram. So it can be more faster to use uniform binary patterns.

Super key frame has shown a very poor performance $20 - 25\%$ accuracy. We've proposed an approach using LTP, which higher order pattern than LBP. Going for further higher order pattern will perform much worse, as higher order pattern would be very sensitive to noise than even LBP.

5.2 Future Work

PCC based shot detection has some drawbacks, like it generates so many shots for gradual transitions. We should utilise some improvised approach for this. LTP dynamic texture doesn't have any sort of temporal information binding so in future that can be improvised with spatio-temporal binding. Shape feature are very powerful for image matching, so fusion of shape moment and LTP dynamic texture can be used in future.

References

[1] T Prathiba and R Kumari. Content based video retrieval system based on multimodal feature grouping by kfcm clustering algorithm to promote human–computer interaction. *Journal of Ambient Intelligence and Humanized Computing*, 12(6):6215–6229, 2021.

[2] B Reddy Mounika, P Palanisamy, Hotta Himanshu Sekhar, and Ashish Khare. Content based video retrieval using dynamic textures. *Multimedia Tools and Applications*, pages 1–32, 2022.

[3] Newton Spolaôr, Huei Diana Lee, Weber Shoity Resende Takaki, Leandro Augusto Ensina, Claudio Saddy Rodrigues Coy, and Feng Chung Wu. A systematic review on content-based video retrieval. *Engineering Applications of Artificial Intelligence*, 90:103557, 2020.

[4] Ajay Kumar Mallick and Susanta Mukhopadhyay. Video retrieval using salient foreground region of motion vector based extracted keyframes and spatial pyramid matching. *Multimedia Tools and Applications*, 79(37):27995–28022, 2020.

[5] YOLOv3-416 object detection model. `Model,Weights:https://pjreddie.com/darknet/yolo/`. Accessed: 2021-11-10.

[6] Zhihu Huang and Jinsong Leng. Analysis of hu's moment invariants on image scaling and rotation. In *2010 2nd international conference on computer engineering and technology*, volume 7, pages V7–476. IEEE, 2010.

[7] Maofu Liu, Yanxiang He, and Bin Ye. Image zernike moments shape feature evaluation based on image reconstruction. *Geo-spatial Information Science*, 10 (3):191–195, 2007.

[8] Reddy Mounika Bommisetty, Om Prakash, and Ashish Khare. Keyframe extraction using pearson correlation coefficient and color moments. *Multimedia Systems*, 26(3):267–299, 2020.

[9] Faiq Baji and Mihai Mocanu. Uniform extended local ternary pattern for content based image retrieval. In *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, pages 391–396. IEEE, 2018.

[10] Jing-Hua Yuan, Hao-Dong Zhu, Yong Gan, and Li Shang. Enhanced local ternary pattern for texture classification. In *International conference on intelligent computing*, pages 443–448. Springer, 2014.

[11] UCF50 Human Action video data set. `Datasetref,UCFImage:https://www.crcv.ucf.edu/data/UCF50.php`. Accessed: 2022-10-15.

[12] Guoping Zhao, Mingyu Zhang, Yaxian Li, Jiajun Liu, Bingqing Zhang, and Ji-Rong Wen. Pyramid regional graph representation learning for content-based video retrieval. *Information Processing & Management*, 58(3):102488, 2021.

[13] Elena Sánchez-Nielsen, Francisco Chávez-Gutiérrez, and Javier Lorenzo-Navarro. A semantic parliamentary multimedia approach for retrieval of video clips with content understanding. *Multimedia Systems*, 25(4):337–354, 2019.

[14] El Mehdi Saoudi and Said Jai-Andaloussi. A distributed content-based video retrieval system for large datasets. *Journal of Big Data*, 8(1):1–26, 2021.

[15] Joey Pinto, Pooja Jain, and Tapan Kumar. A content based image information retrieval and video thumbnail extraction framework using som. *Multimedia Tools and Applications*, 80(11):16683–16709, 2021.

[16] Andre Araujo and Bernd Girod. Large-scale video retrieval using image queries. *IEEE transactions on circuits and systems for video technology*, 28(6):1406–1420, 2017.

[17] N Gayathri and K Mahesh. Improved fuzzy-based svm classification system using feature extraction for video indexing and retrieval. *International Journal of Fuzzy Systems*, 22:1716–1729, 2020.