DEVELOPMENT OF AN IMPROVED KEYFRAME EXTRACTION SCHEME FOR VIDEO SUMMARIZATION BASED ON HISTOGRAM DIFFERENCE AND K-MEANS CLUSTERING

BY

BILYAMIN MUHAMMAD

P17EGCP8061

bsmuhd@gmail.com

A DISSERTATION SUBMITTED TO THE SCHOOL OF POSTGRADUATE STUDIES, AHMADU BELLO UNIVERSITY, ZARIA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A MASTER OF SCIENCE (M.Sc.) DEGREE IN COMPUTER ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

FACULTY OF ENGINEERING

AHMADU BELLO UNIVERSITY, ZARIA

NIGERIA

MARCH, 2021

DECLARATION

I declare that this dissertation titled “Development of an Improved Keyframe Extraction Scheme for Video Summarization Based on Histogram Difference and K-Means Clustering” has been carried out by me in the Department of Computer Engineering, Ahmadu Bello University, Zaria as part of the requirements for the award of the degree of Master of Science in Computer Engineering. The information derived from the literature has been duly acknowledged in the text and a list of references provided. No part of this dissertation was previously presented for another degree or diploma at this or any other institution.

Bilyamin MUHAMMAD ---------------------------- ----------------------
(Student) Signature Date

CERTIFICATION

This dissertation entitled DEVELOPMENT OF AN IMPROVED KEYFRAME EXTRACTION SCHEME FOR VIDEO SUMMARIZATION BASED ON HISTOGRAM DIFFERENCE AND K-MEANS CLUSTERING by Bilyamin MUHAMMAD meets the regulations governing the award of the degree of Master of Science (M.Sc.) in Computer Engineering of the Ahmadu Bello University and is approved for its contribution to knowledge and literary presentation.

Dr. B. O. Sadiq ____________________ ________________
(Chairman, Supervisory Committee) Signature Date

Dr. I. J. Umoh ______________________ _________________
(Member, Supervisory Committee) Signature Date

Prof. M. B. Mu'azu ______________________ ______________
(Head of Department) Signature Date

Prof. Sani Abdullahi ______________________ ______________
(Dean, School of Postgraduate Studies) Signature Date

DEDICATION

This research is dedicated to my Late sisters, Haleematu-Sadiya Muhammad and Asma’ul-Husnah Muhammad. May your gentle souls continue to rest in Jannatul Firdaus.

ACKNOWLEDGMENT

In the name of Allah (SWT), the most beneficent, the most merciful. All praises are due to Allah (SWT). Peace and blessings are upon His Prophet, Muhammad bin Abdullah (SAW), his companions and those that follow their right path until the day of resurrection.

My limitless appreciation goes to my supervisors, Dr. B.O Sadiq and Dr. I. J Umoh for their tireless effort, support, patience, constant guidance and encouragement from the beginning to the end of this work. This would not have been possible without their valuable participation and wonderful corrections towards the research and writing of the dissertation. I feel so lucky and honored having such a wonderful combination of supervisory committee. May the Almighty God continue to bless and reward them abundantly.

My sincere appreciation goes to the Head of the Department, Prof. M. B. Mu'azu for his support, contributions, and encouragement. Thank you so much Sir for the fatherly love and encouragement.

My deep gratitude goes to Dr. Y. Ibrahim, Dr. A. T. Salawudeen, Dr. H. B. Salau, and Mr. O. Ajayi for all the research discussion, motivation and assistance. God bless you Sirs.

I am thankful to all the lecturers of Computer Engineering Department, Ahmadu Bello University, namely; Dr. E. A. Adedokun, Dr. M. B. Abdulrazak, Dr. T. H. Sikiru, Dr. Y. A. Sha'aban, Dr. E. Okafor, Dr. B. Yahaya, Mrs Z. M. Abubakar, Mrs. N. Usman, Mrs. R. Adebiyi, Mr. H. Zaharadeen, Mr. A. Umar, and Mr. A. Abdulfatai. I am highly grateful for all your contributions to my academic career.

To my friends and colleague whom I called my brothers, Abdulhakeem (Don Yogs), Raji Beee, Akan Julius Bello (AJB), Jibril, Habu, Emma, Monday, Momoh Muyidden, Gabi, Femo

Omowuyi, Sman, Maxi, Big Data, AA, Mona, Madam Esther, Modupe, Hudu Magaji, Saliu Shaba, Muhammad Alkali, Sabbah, and those whose names could not be mentioned, I am grateful for the support and all that we shared.

Finally, my deepest gratitude goes to my parents, Dr. S. S. Mohammed and Hajiya Hauwa Gambo for their endless love, financial support, kind advice, understanding and prayers throughout my life thus far. To my siblings; Hafiz, Ahmad, Yusuf, Naja'atu, Mariyah, Khadija, and Rahma, I love you all.

I am really grateful to everyone, for the support and love, May Almighty Allah bless you all.

Bilyamin MUHAMMAD
March, 2021.

ABSTRACT

The rate of increase in multimedia data necessitated the need for a large number of storage devices. Nonetheless, the stored multimedia data has a lot of redundant video frames. These redundant frames make video browsing and retrieval difficult as well as time-consuming for the user; hence, negatively affecting bandwidth utilization and storage capacity. In order to improve the bandwidth utilization and storage capacity, keyframe extraction algorithms were developed. These algorithms were implemented to extract a unique set of frames and eliminate redundant ones. However, despite the achieved improvement in the keyframe extraction process, there exists a significant number of redundant frames in the summarized video. In order to address this issue, this research presents the development of an improved keyframe extraction scheme for video summarization based on histogram difference and k-means clustering. The developed scheme is suitable for the detection of shot transitions and extraction of keyframes in both low motion and fast-moving videos. The histogram-based approach was utilized to detect shot transitions in the video. Furthermore, the k-means clustering approach was used to efficiently extract a unique set of keyframes. The performance of the developed scheme was evaluated on 4 different videos namely; surveillance footage, movie clip, advert, and sport videos which were all obtained from the popular video-sharing website YouTube. Results were compared with existing schemes of Rodriguez *et al.*,(2018) and Sheena and Narayanan (2015) using compression ratio, precision and extraction rates, and f-measure as performance metrics. In terms of the compression ratio, the results showed that the developed scheme outperformed the existing schemes by 24.20% and 35.65%. In terms of precision, it also outperformed the existing schemes by 8.60% and 11.31%. Also, in terms of extraction rate, it outperformed the existing schemes by 0.49% and 7.04%. It also showed an improvement in f-measure by 4.65% and 9.22% when compared with the existing schemes.

TABLE OF CONTENTS

CHAPTER ONE : INTRODUCTION

CHAPTER TWO : LITERATURE REVIEW

CHAPTER THREE : MATERIALS AND METHOD

CHAPTER FOUR : RESULTS AND DISCUSSION

CHAPTER FIVE : SUMMARY, CONCLUSION AND RECOMMENDATION

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

Acronym	**Definition**
BBF	Best Bin First
BoW	Bag of Word
COC	Cumulative Occlusion Curve
CR	Compression Ratio
DST	Dissolve Soft Transition
DWT	Discrete Wavelet Transform
EBA	Edge Based Approach
ECR	Edge Change Ratio
ER	Extraction Rate
FSDWT	Faber Shauder discrete wavelet transform
FST	Fade in/out Soft Transition
HBA	Histogram Based Approach
HCA	Hierarchical Clustering Approach
HDPSC	High Density Peaks Search Clustering
HT	Hard Transition
ISODATA	Iterative Self-Organizing Data Analysis
LDT-SBD	Local Double Threshold Shot Boundary Detection

NNCA	Nearest Neighbor Clustering Approach
PBA	Pixel Based Approach
RGB	Red, Green, and Blue
SBD	Shot Boundary Detection
SIFT	Scale Invariant Feature Transform
ST	Soft Transition
SVD	Singular Value Decomposition
VEP	Video Editing Process
VPP	Video Production Process
WST	Wipe Soft Transition

CHAPTER ONE

INTRODUCTION

1.1 Background to the Research

In recent years, video has become the most widely used multimedia application in the world. However, the massive growth in digital video capturing technologies has resulted to videos being a huge volume of data (Sujatha & Mudenagudi, 2011). For example, the digital surveillance camera deployed for the purpose of public security, can record successive movement of visual frames for a whole day. Hence, resulting in the generation of large amount of feature related frames. As such the storage and transmission of this video becomes difficult as well as time consuming (Asim *et al.*, 2018). To address this issue, an effective video management technology is needed to provide easy access to the video content in lesser time without losing important information (Kumar *et al.*, 2018). This can be achieved by the use of video summarization technology, for example, Keyframe extraction (Asim *et al.*, 2018).

Video summarization also known as video abstraction is the mechanism for eliminating redundant frames and providing a comprehensive view of a full-length video (Santini, 2007). A video comprises of several video shots captured by a single camera at different positions, and these shots are separated by either an abrupt or gradual transition (Del Fabro & Böszörmenyi, 2013). The abrupt transition is a sudden change between successive shots, while gradual transition is a continuous change that occurs between two or more successive shots formed by video editing applications (Lu & Shi, 2013). Among the two types of transitions, detection of gradual transitions has been a major issue for many researchers in the area of video summarization (Abdulhussain *et al.*, 2018). This is due to the fact that gradual transition spans for one or more seconds in videos depending on the editing effects used. There are three types of gradual transition namely; dissolve, fade in/out, and wipe transitions (Abdulhussain *et al.,* 2018). These gradual transitions are mostly applied in fast-moving videos such as movies during the course of production.

The goal of the video summarization is to manage a large amount of video data to make it suitable for browsing, retrieval, and indexing (Li *et al.,* 2017). Dynamic video summarization and keyframe selection are the two approaches for video abstraction (Paul *et al.,* 2018). The dynamic video summarization provides an abstract version of the whole video along with its corresponding soundtrack. While keyframe extraction (also known as static video summarization or representative frames) is an approach that provides a more condensed version of the original video by extracting the representative frames from candidate shots (Gharbi *et al.*, 2016).

Many techniques exist for the detection of shot transitions and extraction of representative frames, so as to reduce the amount of video data to be stored and transmitted over the network. This will in turn improve the bandwidth utilization, storage capacity and also save transmission rate (Azhar *et al.*, 2016). The histogram-based approach also attempts to extract keyframes in a full-length video by computing the mean and standard deviation of the absolute difference between successive video frames (Rodriguez *et al.*, 2018). However, the representative frames extracted are still observed to have feature related frames. These can further be reduced by clustering the similar frames into a single cluster, and select the frames closest to the centroid as keyframes.

1.2 Significance of Research

Video summarization has received a lot of attention from the multimedia industries and academia due to the massive growth of digital video capturing devices and the increasing rate of video transmission over the Internet (Mithlesh & Shukla, 2016). These industries are faced with the problem of bandwidth utilization and storage space as a result of large volumes of the video data (Paul *et al.*, 2018). Due to this fact, the video summarization system is required for real-time applications such as digital surveillance systems. Hence, it requires a keyframe extraction scheme that will accurately extract the set of unique keyframes to represent the entire video and eliminate the redundant frames.

1.3 Statement of Problem

Video data storage and transmission over the Internet has become difficult because of the huge amount of visual contents present in the video (Asim *et al.,* 2018). A keyframe video summarization approach has been proposed by researchers such as Rodriguez *et al.,* (2018) to make video browsing and retrieval easy for the User. However, frames generated due to the presence of gradual transitions, camera zooming, and sudden illuminance in the video can never guarantee optimal utilization of bandwidth and storage. Therefore, there is a need to develop an improved scheme to eliminate these redundant frames to achieve a better compression ratio and reduce the time at which the video files are being retrieved. Given this, this research developed the use of a histogram-based approach for shot boundary detection and k-means clustering for the unique set of keyframes extraction. This is necessary to improve the storage capacity and transmission rate while eliminating the redundant frames extracted.

1.4 Aim and Objectives

This research aims to develop an improved keyframe extraction scheme for video summarization based on histogram difference and k-means clustering.

To achieve the aim, the following objectives are set.

i. To implement a shot boundary detection scheme based on a histogram difference between consecutive video frames.

ii. To develop a keyframe extraction scheme based on k-means clustering.

iii. To evaluate and compare the proposed technique with that of Rodriguez *et al.* (2018) and Sheena and Narayanan (2015) based on compression ratio, precision and extraction rates, and f-measure.

1.5 Dissertation Organization

Chapter one presents the general introduction of this work. The rest of the chapters are structured as follows: Chapter two gives details of the reviews of related literature and relevant fundamental concepts about the video, video hierarchy, video summarization, hard and soft video transition. Chapter three presents the methods for developing the improved keyframe extraction scheme for video summarization based on histogram difference and k-means clustering. The analysis, performance, and discussion of the results are given in Chapter four. Chapter five comprises the conclusion and recommendation for future work. At the end of the dissertation, the list of the cited references and MATLAB codes are provided in the appendices.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This section comprises of two sub-sections. First, the fundamental concepts related to the subject matter is discussed. Second, similar works related to this work are reviewed.

2.2 Review of Fundamental Concepts

In this section, the fundamental concepts such as video, video hierarchy, video summarization, hard and soft video transition are discussed.

2.2.1 Video

A video is a visual data that comprises of multiple images called frames with a specific frame rate accompanied by soundtracks (Adedokun *et al.,* 2019). The frame rate is computed in frame per second (fps)**.** These video frames are individual pictures in a sequence of images. A one second (1s) video may contain about twenty-five to thirty frames having similar visual content and the same size. The time sequence between two successive frames is equal, typically 1/25 or 1/30 seconds. A video is also a 3D signal in which the vertical axis represents the frame height while the horizontal axis is the frame width representing the visual content of the video, whereas the third axis represents the difference in the frame along with time. (Amiri & Fathy, 2010; Yuan *et al.,* 2007). Figure 2.1 gives an illustration of a video signal sample.

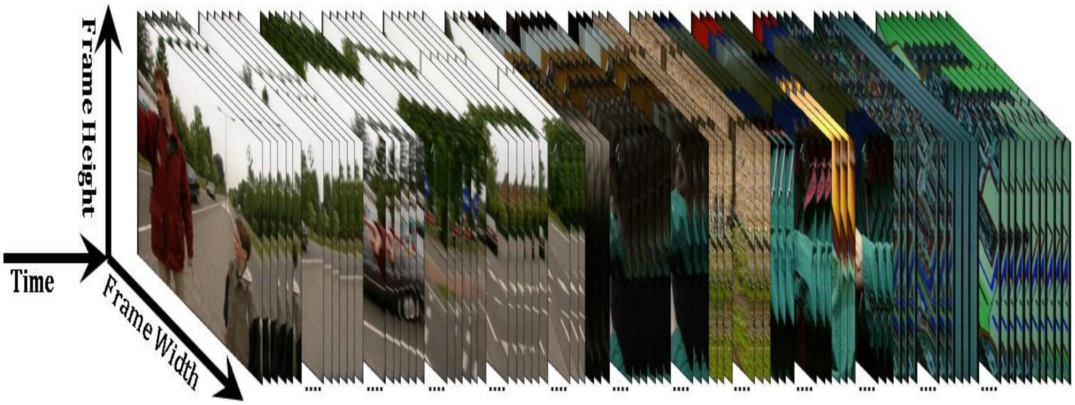


Figure 2.1: Video Signal Sample (Abdulhussain *et al.*, 2018).

2.2.2 Video Hierarchy

A video hierarchy is the entire structure of a video that comprises of scenes, shots, and frames. For instance, it is closely related to a textbook consisting of several chapters known as a single story or multiple stories in the video (Bhaumik *et al.*, 2016). A story comprises of a number of scenes that captures the sequence of event. Hence it is made up of interrelated shots recorded at different camera positions (Liu *et al.*, 2013). Figure 2.2 gives an illustration of a video hierarchy. A shot is the smallest unit of temporal visual information (Lu & Shi, 2013). It contains a sequence of interrelated frames captured uninterruptedly by a single camera (Del Fabro & Böszörmenyi, 2013). These frames represent certain related actions or events in time and space (Birinci & Kiranyaz, 2014; Janwe & Bhoyar, 2013). The intra-shot images consist of related information and visual contents with temporal differences (Chen *et al.,* 2010; Tong *et al.*, 2015). These differences in time between frames may cause small or large changes as a result of actions between the start and stop marks (Asghar *et al.*, 2014). Hence, the scene and shot structures of a video are analogous to a sentence and paragraph. Shots are important in depicting a story. While scenes for showing visual narrative (Liu *et al.*, 2013).

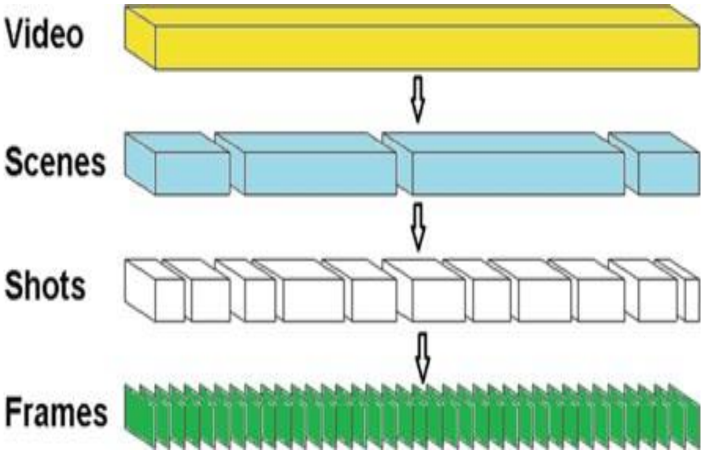


Figure 2.2: Video Hierarchy (Rodriguez *et al.*, 2018).

2.2.3 Video Transition

The transition can be defined as a frontier between multiple video shots (Abdulhussain *et al.*, 2018). The Video Editing Process (VEP) is employed to merge multiple shots to generate a video during the Video Production Process (VPP) (Küçüktunç *et al.*, 2010). These VEPs allows the generation of various transition effects. The main types of shot transitions are shown in Figure 2.3.

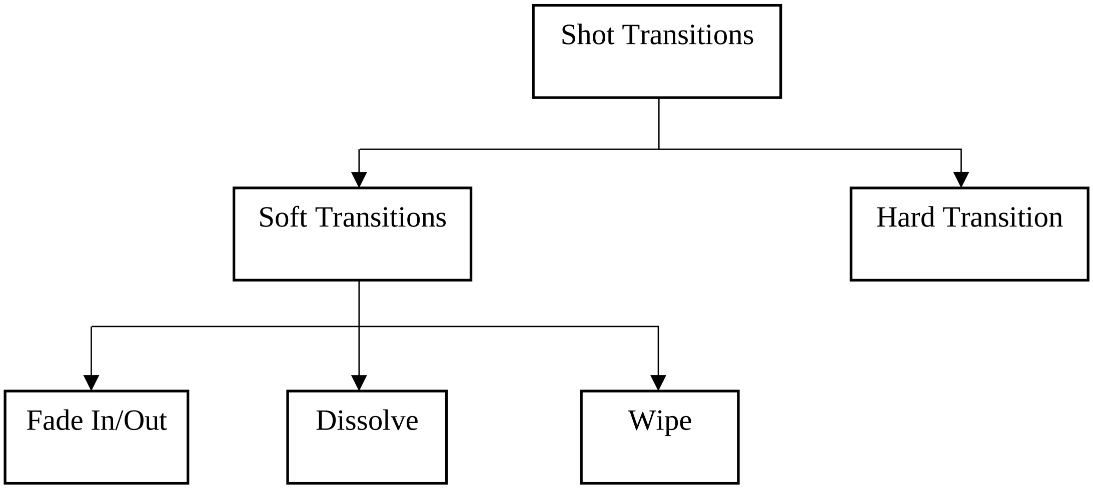


Figure 2.3: Types of Shot Transition (Zedan *et al.*, 2018)

2.2.3.1 Hard Transition

Hard Transition (HT) (also known as a cut or sharp transition) is an abrupt change which occurs when there is a sudden change between two successive video shot without any VEP (Ling *et al.*, 2008). Thus, it can be concluded that hard transition occurs between the last frame of a shot and the first frame of the following shot. Figure 2.4 shows a sudden change between the last frame of the current shot and the first frame of the subsequent shot (i.e., between frames 3 and 4).

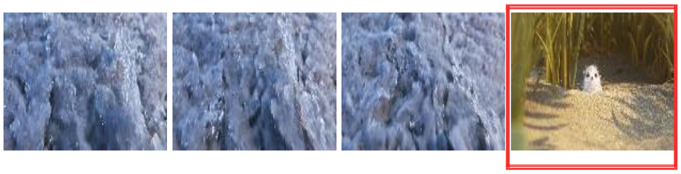


Figure 2.4: Hard Transition (Bi *et al.*, 2018).

2.2.3.2 Soft Transition

Soft Transition (ST) is also known as continuous transition or gradual transition (Lu & Shi, 2013). ST occurs when two successive shots are combined by making use of the video editing process throughout the course of a production. It may span two or more video frames that contain truncated information and are visually interdependent (Jiang *et al.*, 2013). In detecting shot boundary or extraction of representative frames from a given video file containing soft transition, the result of the operation might not be efficiently achieved. This is because of the high visual content similarities between the consecutive frames involved in the VEP (Abdulhussain *et al.*, 2018). There are several types of soft transition namely, dissolve, wipe, and fade in/out.

i. **Dissolve Soft Transition (DST):** DST is the process in which the pixel intensity values gradually diminish from the current shot, and the values of the pixel intensity of the next shot gradually appear (Choroś, 2011). In DST, two or more frames may have different pixel intensity values but contains the same visual information as shown in Figure 2.5.

Figure 2.5 depicts only one frame (i.e., 1759th frame) that is utilized in the dissolve transition.

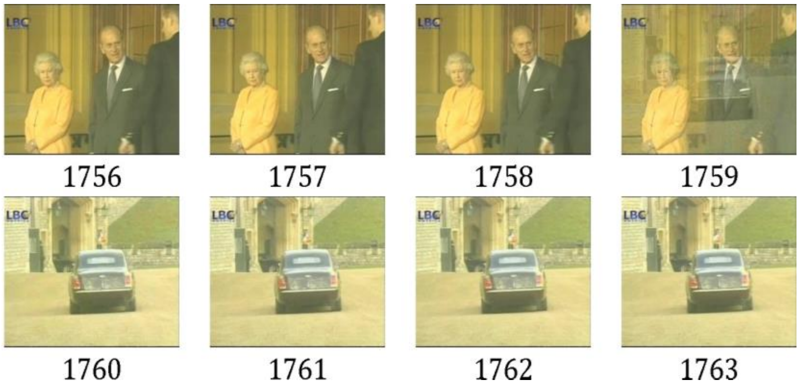


Figure 2.5: Dissolve Transition (Abdulhussain *et al.*, 2018)

ii. **Fade in/out Soft Transition (FST):** FST is the type of transition that is usually applied in movies to start a scene smoothly. In fade-in transition, one or more end frames of the shot are directly changed by a fixed intensity frame, and the pixel intensity values of the next shot gradually appear into position from a completely dark sequence (Cernekova *et al.*, 2005). Figure 2.6 shows an example of a fade-in transition with 31 frames involved in the transition (n = 1-32)

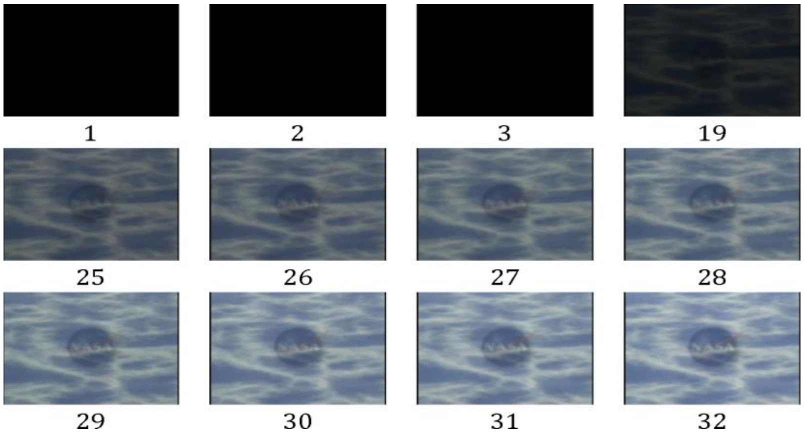


Figure 2.6: Fade-in Transition (Abdulhussain *et al.*, 2018).

In fade-out transition, only frames at the end of the current shot are involved in the transition process, with no frames from the next shot are involved in the transition process. It is usually applied at the end of a movie scene. Figure 2.7 illustrates a fade-out transition involving 13 frames (n = 28,735–28,750)

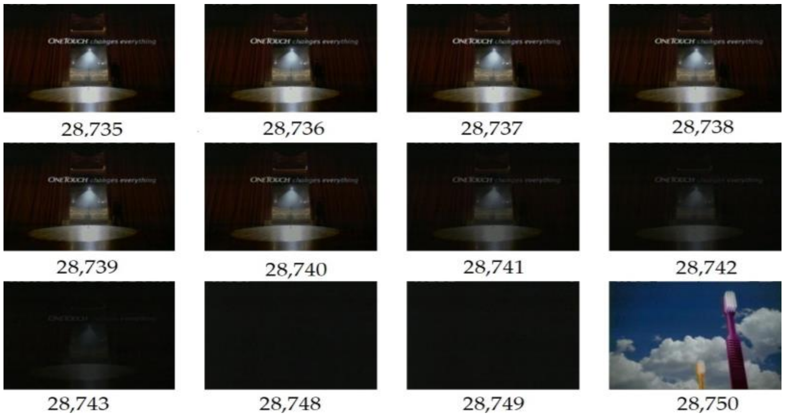


Figure 2.7: Fade-out Transition (Abdulhussain *et al.,* 2018).

iii. **Wipe Soft Transition (WST)**: WST is the process in which the current shot pixels are progressively superseded by the corresponding pixels from the next shot by following an organized spatial pattern (Kawai *et al.,* 2007). Figure 2.8 shows the gradual substitution of the column pixels from left to right of the frame.

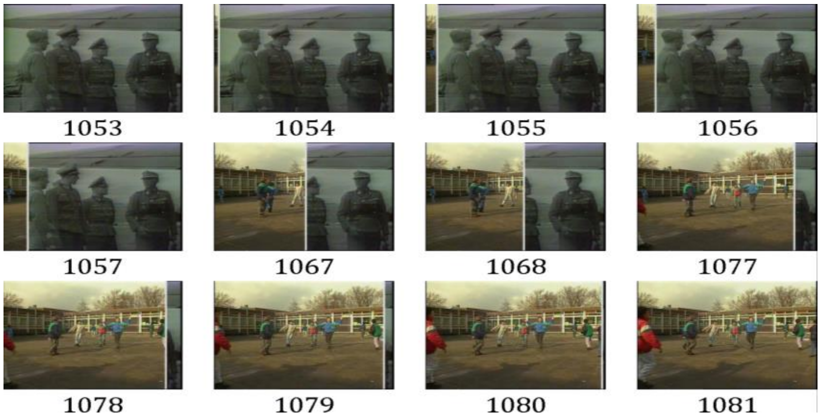


Figure 2.8: Wipe Transition (Abdulhussain *et al.*, 2018).

2.2.4 Video Summarization

Video summarization is defined as the process of presenting an abstract view and comprehensible analysis of a full-length video within the shortest period of time while preserving the essential activities of the original video (Santini, 2007). The rapid growth in network infrastructure together with the use of advanced digital video technologies reveal the need for video summarization technologies to manage the huge volumes of video data generated by enormous multimedia applications (Kumar et al., 2018). Hence, allowing the users to access and retrieve the relevant contents of the video easily without viewing the entire video. Some of the areas touched by the development of the video summarization techniques are; e-learning, news broadcast, home videos, sports, movies among other areas (Furini *et al.,* 2010).

Figure 2.9 shows a block diagram of a video summarization technology. The system consists of the shot boundary detection module, where the video frames are partitioned into number of shots and the keyframe extraction module where the number of representative frames are identified as well as selected in order to provide a summarized video.

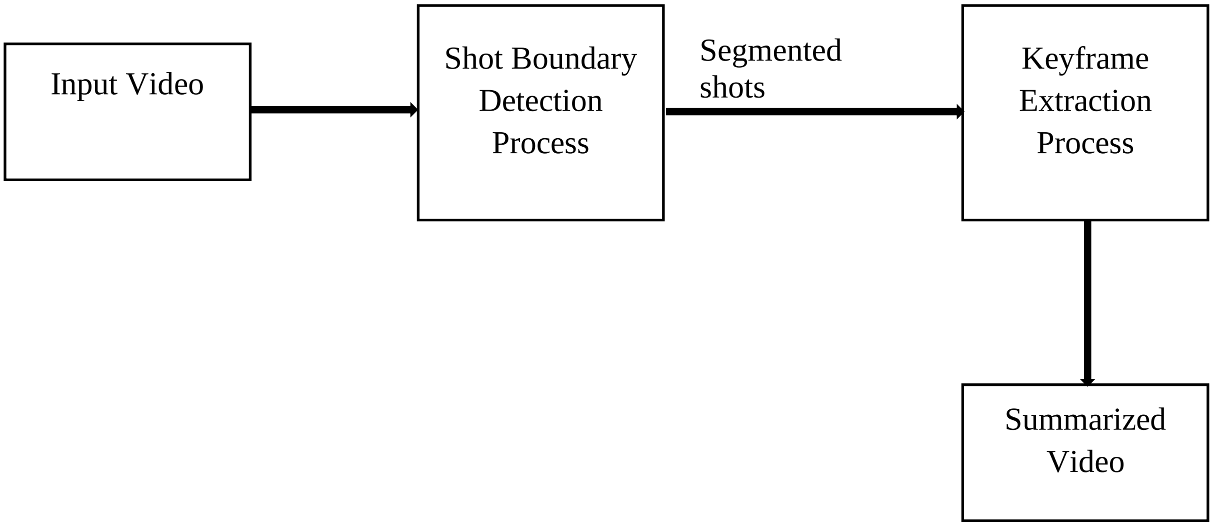


Figure 2.9: Block Diagram of Video Summarization Technology (Furini *et al.*, 2010)

2.2.5 Shot Boundary Detection

Shot Boundary Detection (SBD) is the process of segmenting a whole video file into various shots (Priya & Domnic, 2014). These shots consist of related video frames having similar visual contents captured by a single camera, and are separated by a boundary. The SBD system consists of three stages. At the first stage, the total number of frames in a given video are extracted. Secondly, the variation between successive frames is computed. Finally, a comparison is established between the frames' difference and a predefined threshold value, and a shot boundary is detected if the difference between the successive frames is greater than the threshold value. Figure 2.9 shows a typical Shot Boundary Detection (SBD) system.

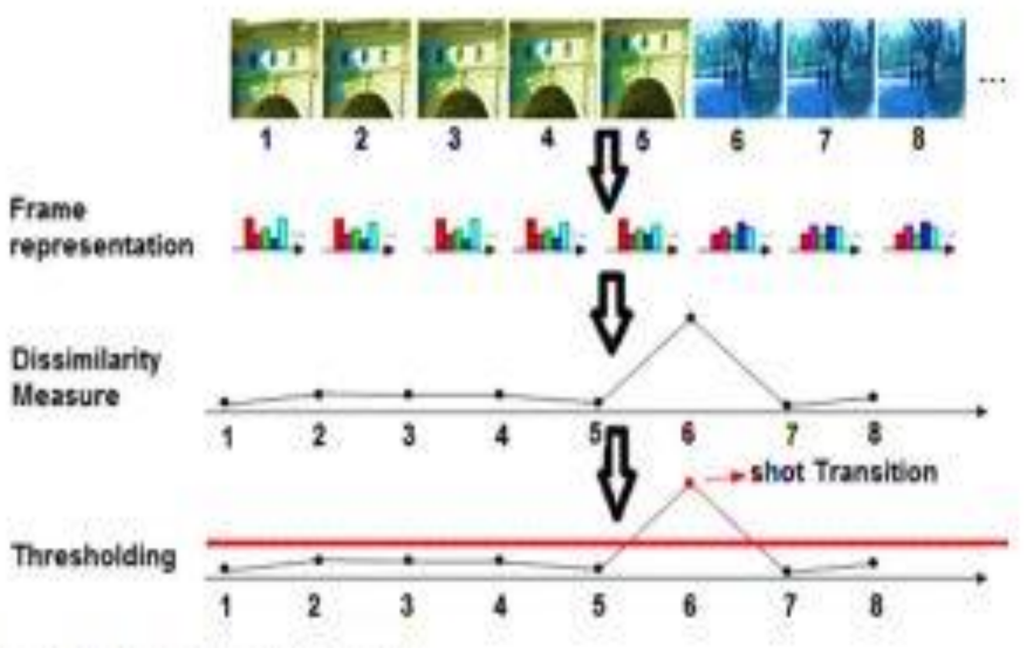


Figure 2.10: Typical Shot Boundary Detection System (Zedan *et al.*, 2018)

Basically, there are three major SBD methods namely; pixel-based, histogram-based, and edge-based methods (Abdulhussain *et al.*, 2018).

2.2.5.1 Pixel Based Approach (PBA)

In this type of SBD method, the difference between two successive images is determined by comparing their pixel values using equation 2.1. The total sum of these pixel differences is calculated and a comparison with a threshold is established. PBA is responsive to flashlight, camera movement, and it is computationally complex (Wu, 2011).

$$\frac{\left|\sum_{i=1}^{R}\sum_{j=1}^{C} I(n,i,j)-\sum_{i=1}^{R}\sum_{j=1}^{C} I(n-1,i,j)\right|}{256\cdot R\cdot C} > \tau \qquad (2.1)$$

where I (n, i, j) and I (n-1, i, j) is the intensity value of the current and previous images in pixel (i, j), R is row, C is column, and τ is the threshold value. A transition is detected if the sum of the difference exceeds the threshold value (Lee *et al.*, 2001).

2.2.5.2 Edge Based Approach (EBA)

This type of technique efficiently detects a boundary when the positions of edges of the current frame show a huge difference with that of the next frame. The Edge Change Ratio (ECR) is utilized to find the edge changes using equation 2.2. In EBA, transitions are detected by looking for a large edge ratio (Wu, 2011). Although, EBAs detect abrupt transition more accurately than the histogram-based approach; however, they are less reliable compared to the Histogram Based Approach (HBA) in terms of performance and computational time (Dailianas *et al.*, 1996). There are several types of EBAs namely; Roberts, Sobel, Prewitt, Laplacian of gaussian, and canny edge detection techniques (Olaniyi, 2014).

$$ECR_n = max\left(\frac{X_n^{in}}{\sigma_n}, \frac{X_{n-1}^{out}}{\sigma_{n-1}}\right) \qquad (2.2)$$

where $\sigma_n$is the number of edge pixels in the frame, $\sigma_{n-1}$is the number of edge pixels in the previous frame, X_n^{in} is the number of edge pixels the current entering, and X_{n-1}^{out} is the number of edge pixels leaving two successive images.

2.2.5.3 Histogram Based Approach (HBA)

An image histogram is the graphical representation of the pixel intensity level (represented along the x-axis) with respect to the number of its occurrence (represented on the y-axis) in an image. It is considered as a replacement for the PBA due to its utilization of the temporal information between two consecutive frames rather than the spatial information (Tapu *et al.*, 2011). The HBA involves computing the histogram difference of consecutive frames and finding the threshold value by calculating the mean and standard deviation of the histogram differences (Wu, 2011). The histogram difference between successive frames is computed using the following (Kathiriya *et al.*, 2013).

$$R(i, i+1) = \sum_{a=1}^{3} \frac{[H(i,a) - H(i+1,a)]^2}{H(i,a)} \tag{2.3}$$

The mean (μ) and standard deviation (σ) of the histogram differences are computed using the following (Kathiriya *et al.*, 2013).

$$\mu = \sum_{i=1}^{M-1} \frac{R(i,i+1)}{M-1} \tag{2.4}$$

$$\sigma = \sqrt{\sum_{i=1}^{M-1} \frac{[R(i,i+1) - \mu]^2}{M-1}} \tag{2.5}$$

The threshold value is determined using the following (Kathiriya *et al.*, 2013).

$$\tau = \mu + \sigma \times C \tag{2.6}$$

Finally, a shot transition is detected if the histogram difference between the successive frames is greater than the threshold value using the following (Wu, 2011).

$$\sum_{a=1}^{3} \frac{[H(i,a)-H(i+1,a)]^2}{H(i,a)} > \tau \qquad (2.7)$$

where $R(i, i+1)$ is the histogram difference. *i^{th}* and *$(i+1)^{th}$* are the current and next frames. *H(i, a)* and *H(i+1, a)* are the histogram of the color channels for consecutive frames. *M* is the total number of frames. *C* is pre-specified constant. This approach has low a computational cost. However, two images having similar histograms but different visual contents will be missed during detection (Tapu *et al.,* 2011).

2.2.6 Keyframe Extraction

Keyframe extraction is an efficient method used to clearly express the important contents of a video file by extracting a set of representative frames and removing/deleting the duplicated ones from the original video (Paul *et al.,* 2018). These extracted keyframes are expected to represent and provide comprehensive visual information of the whole video (Gharbi *et al.,* 2016). The keyframe approach is employed to reduce the computational burden and the amount of data needed for video processing as to make indexing, retrieval, storage organization, and recognition of video data more convenient and efficient (Sheena & Narayanan, 2015). These techniques can be classified into three main classes, namely: shot based, sampling-based, and clustering-based techniques (Asim *et al.,* 2018).

2.2.6.1 Sampling-Based Technique

This is a type of method that selects representative frames by uniformly or randomly sampling the video frames from the original video, without giving importance to the video content (Asim *et al.,* 2018). The concept of this technique is to choose every kth frame from the original video. This value of k is determined by the duration of the video. A usual choice of duration for a summarized video is 5% to 15% of the whole video. For the case of 5% summarization, every 20th frame is selected as the keyframe, while for the case of 15% summarization, every 7th frame is selected as the keyframe (Jadon & Jasim, 2019). These keyframes extracted do not represent all the content of the original video, and may also result in redundant frames having similar contents (Tirupathamma, 2017).

2.2.6.2 Shot Based Technique

In this approach, an efficient SBD method that detects shot boundary/transition is utilized first. After segmenting the video frames into various shots, the keyframe extraction process is then performed. Different kinds of literature have discussed different techniques for the selection of key frame. The traditional approach is to select the first and last frames of the candidate shot as the key frames (Tirupathamma, 2017). These extracted key frames are the representative frames of the shots, which in turn produces the summary of the original video in a more condensed manner (Kaur & Kumar, 2018).

2.2.6.3 Clustering Based Technique

Clustering is an unsupervised learning approach that finds sets of similar data points and cluster them together. In this method, frames within a video file having similar visual contents are partitioned into different number of clusters. From each cluster, the frame that is nearest to the center of the candidate cluster is extracted as key frame (Paul *et al.*, 2018). The frame similarities are determined by the features they exhibit such as color histograms, texture, saliency maps, and motion (Li *et al.*, 2017). The most common and widely used clustering-based techniques are: Hierarchical clustering and K-means clustering algorithm (Janwe & Bhoyar, 2016).

i. **Hierarchical Clustering Algorithm (HCA):** It is a clustering approach that group similar entities based on hierarchy. In this method, the clusters are usually arranged in the form of a tree diagram where multiple clusters that are near to each other are merged into a single cluster. This process continues till all the clusters in the dataset becomes one (Coates et al., 2011). There are two main types of HCA namely; Agglomerative and divisive hierarchical clustering.

   In the agglomerative hierarchical clustering, the clusters are built bottom-up beginning with different data points and finishing with only one group. While in the divisive hierarchical clustering, a single cluster in broken into different data points. These hierarchical clustering approaches have a high computational time and requires large number of memory space. Thus, limiting its application to relatively small datasets (Coates et al., 2011).

ii. **K-Means Clustering Algorithm**: is an iterative approach that partitions N entities into K sets such that each entity goes to the cluster with the nearest mean. The clustering approach begins by randomly selecting K data points as initial centroids. These centroids are the mean of all the data points that belong to the clusters. The distance between each centroid and every data point is then calculated**.** Each data point is allocated to the cluster with minimum distance. After assigning every data point to the clusters, the centroids are re-calculated using the data points of the newly formed clusters as shown in equation 2.8 (Hu *et al.*, 2008).

$$J = \sum_{j-1}^{k} \sum_{i-1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{2.8}$$

where $\left\| \boldsymbol{x}_{\boldsymbol{i}}^{(\boldsymbol{j})} - \boldsymbol{c}_{\boldsymbol{j}} \right\|^{\boldsymbol{2}}$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the n data points from their respective cluster centres.

The advantages of this clustering approach over the HCA is that it is simple, interpretable, and applicable on large datasets (Huang & Wang, 2018). This makes it suitable for the research as it deals with large number of video frames. The video frames with similar visual features are clustered into a single cluster, and the frame(s) that are closest to the centroid are extracted as the keyframe(s) (Paul *et al.,* 2018). Figure 2.11 shows how keyframes are extracted using K-means clustering approach.

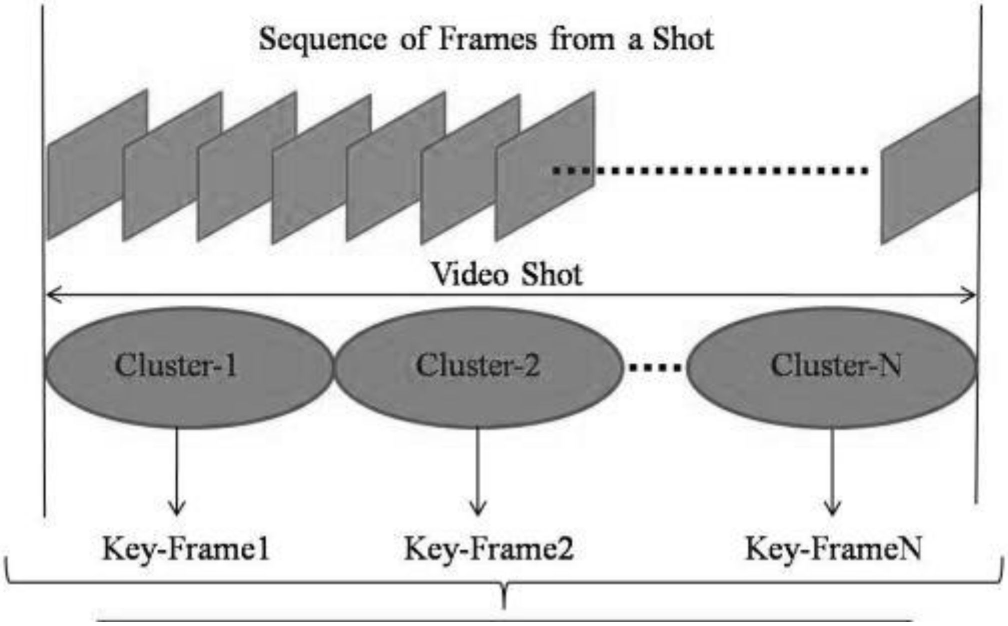


Figure 2.11: K-Means Clustering Approach (Paul *et al.,* 2018).

2.2.7 Evaluation Metrics

To test the performance of the developed key frame extraction technique, several evaluation metrics are utilized namely; compression ratio, precision and extraction rates, and F-measure (Sheena & Narayanan, 2015).

i. Compression Ratio

The Compression Ratio (CR) is employed to measure the compactness of the technique due to the extracted key frames. CR is calculated using the following (Gharbi *et al.,* 2016).

$$CR = \left\{1 - \frac{N_k}{N_f}\right\} \times 100\% \tag{2.9}$$

where N_f is the total number of frames in the original video. N_k is the total number of the extracted key frames.

ii. Precision and Extraction Rates

Precision rate also known as positive predictive value (Murugan *et al.*, 2018). It is defined as the ratio of the total number of key frames extracted accurately (N_a) to the total number of key frames extracted by the technique from the original video (N_k). In other words, precision is the process of measuring the accuracy of the key frame extraction technique, and computed using the following (Paul *et al.*, 2018).

$$Precision = \frac{N_a}{N_k} \times 100\% \tag{2.10}$$

Extraction Rate (ER) is defined as the total number of keyframes extracted accurately divided by the total number of ground truth frames. It is computed as follows (Murugan *et al.*, 2018).

$$ER = \frac{N_a}{N_a+N_m} \times 100\% \tag{2.11}$$

where N_a is the number of frames extracted accurately. N_k is the total number of the extracted key frames from the original video. N_m is the number of missed key frames from the video frames.

iii. F-measure

F-measure also known as F-score is the method of evaluating the performance of an algorithm by merging multiple evaluation metrics to obtain one metric using the Harmonic mean. F-score is computed using the following (Abdulhussain *et al.,* 2018)**.**

$$F = 2 \times \frac{Precision \times ER}{Precision + ER} \qquad (2.12$$

2.3 Review of Similar Work

In this section, related works that have been carried out in the area of keyframe extraction are reviewed. The relevance of this review is to provide an insight to different types of techniques utilized in the research area. The knowledge gained enabled the developed scheme by employing different approach to get an improved result.

Liu *et al.*, (2009) presented a method for detecting shot transitions and selecting key frames using Scale Invariant Feature Transform (SIFT). The proposed technique was implemented in three stages. Firstly, the authors adopted the SIFT to find the variation between the visual features of consecutive frames. Secondly, a novel approach called Local Double Threshold Shot Boundary Detection (LDT-SBD) was then implemented to address the issue of false SBD caused by SIFT key points, and also segment the video frames into shots. Lastly, Best Bin First (BBF) technique was employed to extract the representative frames. The BBF was utilized to match the SIFT key points between consecutive images in the candidate shot. The images with most SIFT key points were then selected as the representative frames. The experimental results indicated that the proposed novel technique can detect both abrupt and gradual transitions efficiently. However, a number of frames affected by sudden illuminance were extracted as keyframes resulting to more redundant frames in the summarized video. In addition, this work has a high computational time due to the utilization of the three modules during the extraction process.

Ren *et al.*, (2010) proposed a method for selecting keyframes using frame information entropy and edge matching rate. The authors first extracted the total frames from the input video. The information entropy for each frame was then computed. The authors also adopted the Prewitt operator to extract and match the edges of the successive frames. If the edge matching rate is up to 50%, the current frame is considered redundant and hence, eliminated. Although the approach can extract a set of unique keyframes, it failed in the detection of frames affected by flashlights. Hence, extracting multiple feature related frames with different illumination intensity as keyframes. In addition, this work has a high computational time due to the utilization of the edge-based approach.

Cao *et al.*, (2012) presented an approach for extracting keyframes using color features. In this approach, the authors considered the first image in the video as the reference image and segmented the remaining images into blocks. The color mean variations between corresponding blocks in the reference and current image were then calculated. The varying blocks in the current images in relation to the varying blocks in the reference image were then counted. If the counted number was more than a predefined threshold, then the current image was selected as keyframe. The experimental result showed that the proposed method could detect camera movement efficiently, and extracts keyframes in both abrupt and gradual video shots. However, this work can only select keyframes in videos having a high variation in color intensity between the frames.

Chao *et al.*, (2012) implemented an augmented 3D based approach for the selection of representative frames in a surveillance video. The video frames were first extracted and segmented into number of shots using any of the shot boundary detection approaches. A keyframe is then selected from each video shot. The authors then integrated the proposed approach with a user interface in order to provide an interactive video retrieval and browsing scheme. The experimental results showed that the proposed approach provided a condensed and meaningful summarized video at the output. However, this work is only suitable for low motion video where the camera position is fixed.

Ejaz *et al.*, (2012) proposed a novel approach for the selection of representative frames based on aggregation mechanism. The authors used the relationship of RGB color channel, color histogram, and moment of inertia to measure the dissimilarities between consecutive video frames. The aggregation mechanism was then employed to merge these measures to select the representative images. Also, the authors adopted the Euclidean distance to filter out similar images from the set of representative images extracted. The representative images with more than 50% similarities were considered redundant images and were eliminated. The experimental results showed that the proposed technique could extract a unique set of key frames but could not differentiate two similar frames having different illumination intensity. Hence, resulting in extraction of similar frames as keyframes.

Wang *et al.*, (2012) presented a new technique for selecting representative frames using Cumulative Occlusion Curve (COC) for semi-automatic two-dimensional to three-dimensional (2D-3D) system. The algorithm was implemented in two main stages. The first stage was the segmentation of the video frames into a set of shots using histogram-based approach. A principle in 2D-3D system was then employed to filter out images between two soft boundary shots. The second stage was the extraction of representative images from the segmented shots using cumulative occlusion curve. The occlusion between successive images was determined using stereo correspondence approach. If the occlusion space in the current image is more than that of the previous image, then the current image is selected as a representative image. The experimental result obtained showed that the algorithm could efficiently extract keyframes from videos with gradual transitions. However, multiple feature related frames affected by the present of flashlights are extracted as keyframes. Hence, resulting to more redundant frames in the summarized video.

Zhang *et al.*, (2013) implemented a new approach for extracting key frames based on modified Iterative Self-Organizing Data Analysis (ISODATA**)** for use with motion capture data. The algorithm focused on two main aspects. Firstly, the similarity distance between consecutive images was utilized to group the motion sequences into two categories. After that, an adaptive threshold needed for the clustering stage was computed. Secondly, the modified ISODATA was employed to cluster all frames. The frame closest to the centroid of each group was automatically selected as representative frame of the sequence. The experimental results demonstrated that the approach can summarize motion capture data efficiently. However, the proposed approach has a high computational time and can also extract gradual transitioned frames as key frames.

Azeroual *et al.*, (2014) proposed a new technique for selecting representative frames using Faber Shauder Discrete Wavelet Transform (FSDWT). The input video frames were first converted to gray scale. The dominant blocks representing the video frames were then computed to generate feature matrices using FSDWT. The authors also adopted a sliding window Singular Value Decomposition (SVD) to determine the rank of these matrices. The calculated ranks were then traced to detect the beginning and ending of video shots. if an image has the highest rank between two successive video shots, then that image is selected as a representative frame. From the experimental results obtained, the proposed novel technique can detect all types of gradual transitions efficiently. However, two similar frames having different illumination intensity are selected as keyframes. Hence, resulting in extraction of similar frames as keyframes.

Raikwar *et al.,* (2014) proposed a method for selecting keyframes based on human assumption. The total video frames were first extracted and stored in a predefined location. The authors then directly select the first image in every shot as the representative image. Although the proposed method can select keyframes at a very low processing time. However, the extracted keyframes might not necessary be the most representative images of the original video. Hence, resulting to missed detection.

Yuan *et al.,* (2014) implemented a method for extracting representative frames from vehicle surveillance video based on AdaBoost classifier. The algorithm was implemented in two modules. The first module involved training the AdaBoost classifier to select the region and integral channel features as the frame feature descriptors. The second module involved utilizing the trained AdaBoost classifier to select the representative images. From the experimental result obtained, the proposed approach can extract unique set of representative frames with less transitioned frames. However, this work has a high computational time because of the well-trained model needed.

Benni *et al.*, (2015) presented a novel approach for shot transition detection and selection of representative images using Eigen values. First, a data matrix was created for all the successive frames in the original video. Covariance matrix was then calculated to determine the dissimilarities between the intensity levels of successive images. A modified approach for calculating the covariance matrix was also presented to reduce the computational cost of recalculating the whole matrix whenever a new image is added to the data matrix. The calculated covariance matrix was then utilized to determine the Eigen values. The minimum Eigen value selected was utilized to determine the variations between the frames. A comparison was established between the minimum Eigen value and a predefined threshold. If the eigen value exceeds the threshold, then the previous image is considered as a transition point and the current image is selected as the representative frame. From the experimental result obtained, the proposed approach can extract keyframes in a video containing hard transitions efficiently. However, the algorithm failed to detect gradual transition leading to the extraction of redundant keyframes.

Jadhava and Jadhav, (2015) presented a technique for extracting keyframes based on higher order color moments. The video frames were first partitioned into M X N block. Then each block is divided into shots using frame histogram, skew and kurtosis values. From each shot, frames with most mean and standard deviation values are selected as the representative frames. The experimental result shows that the technique can extract set of keyframes with less wipes effects. However, it failed in the detection of fade in/out gradual transition resulting to the extraction of redundant keyframes.

Kavitha and Rani, (2015) implemented a technique for selecting keyframes based on prioritized fusion approach. The authors utilized both Discrete Wavelet Transform (DWT) and static attention methods to select the keyframes from low and high motion videos. The proposed technique is implemented in two main stages. The frames from the given video were first extracted, and partitioned into number of shots using the sobel edge detection approach. From each video shot, a keyframe is then selected based on the static attention and discrete wavelet transform methods. Although, the proposed approach extracts keyframes from videos with abrupt transition accurately, it failed in the detection of any form of the gradual transitions leading to the selection of redundant frames. In addition, the proposed approach has a high computational time due to the utilization of edge-based approach for the detection of video transition.

Sheena and Narayanan, (2015) proposed a statistical based approach for the extraction of keyframes for video summarization. The algorithm was implemented in two modules. Firstly, the video frames were extracted and the histogram of the individual frames were computed. secondly, the mean and standard deviation of the absolute difference of the frame's histogram were computed. A comparison was then established with a predefined threshold. If the difference between consecutive frames is higher than the threshold, then a keyframe is extracted. The experimental results demonstrated that the proposed technique is computationally simple and can efficiently extract set of key frames from a video with abrupt shots. However, it failed in the detection of gradual transitions present in the original video leading to the extraction of redundant frames.

Guo *et al.*, (2016) proposed an approach the extraction of keyframes using relative entropy and extreme Studentized deviate test. The proposed approach was implemented in two main modules. Firstly, the video frames were extracted, and the distance between successive frames were computed. The computation was carried using the relative entropy and its square root. Secondly, the extreme studentized deviate test was utilized to determine the transitions between frames in order to partition them into shots. If the variation in visual content is much, then the candidate shot is segmented into number of sub-shots. A keyframe is then selected from the sub-shot. From the experimental result obtained, the proposed approach can extract keyframes in a video containing hard and soft transitions efficiently. However, the algorithm extracts multiple feature related frames from different video shots leading to the extraction of redundant keyframes.

Janwe and Bhoyar, (2016) presented a new approach for the selection of representative images using unsupervised clustering technique and mutual comparison. They employed an unsupervised clustering method to extract the key frames. Once the key images were selected, a mutual comparison to find the similarities between two successive key frames was carried out. This mutual comparison was utilized to filter out any duplicated key frame having similar visual contents with its neighboring key frame in a particular shot. The new approach is computationally simple and can filter out duplicated images in a shot efficiently. However, it failed in the detection of gradual transitions leading to the selection of redundant frames

Rashmi and Nagendraswamy, (2016) implemented an approach for detecting abrupt shot transition and selection of representative frames using bitwise exclusive or (XOR) logical operation. The images from the original video were first transformed into gray scale, and each pixel value in the gray scaled images was represented in its corresponding binary form (i.e. 0s and 1s). The XOR operation was then utilized at the pixel locations of two consecutive images. A shot transition is detected if the dissimilarity between two successive images exceeds a given threshold. The author further used the bitwise XOR variation technique on each of the segmented shots to create a variation matrix. Using the variation matrix obtained, a key frame was then extracted. Although, the proposed technique was easy to implement and can detect abrupt shot transition effectively, it failed in the detection of any form of the gradual transitions leading to the selection of redundant frames. In addition, the proposed approach has a high computational time due to the number of logical operations performed on each of the pixel values of the video frames.

Gharbi *et al.*, (2017) implemented a graph modularity clustering-based approach for the extraction of keyframes. In this technique, the frames in the original video were first extracted. A SURF detector and repeatability table were then utilized to determine the similarities between consecutive video frames. The authors also utilized the windows rule technique to decrease processing time of extraction. The experimental result showed that the proposed approach can extract representative frames at low computational time. However, feature related frames with different illumination intensity were extracted as keyframes. Hence, resulting to more redundant frames in the summarized video.

Li *et al.*, (2017) implemented a novel technique to select representative frames in high dimensional space called the summary space. The proposed technique is implemented in three main stages. Firstly, the frames from the original video were extracted and mapped to the summary space by a Lipschitz smooth real function. An unsupervised clustering approach was then performed on the video frames in the high dimensional space. This clustering algorithm was employed to extract the key frames in the summary space. Secondly, a perceptual hashing approach was utilized to determine the similarities between the key frames. Finally, the similar frames were filtered out, and a more comprehensive key frames were obtained. The experimental result shows that the technique can extract set of non-redundant and unique representative frames in videos with abrupt transitions only. However, this work has a high computational time due to the processes involved in the extraction of the representative frames.

Li *et al.*, (2017) presented a keyframe extraction scheme based on sparse coding. The video frames were first extracted and segmented into number of shots using the dictionary items created by the sparse coding. The similarities between consecutive frames were then computed. Finally, frames higher features are selected as keyframes. The experimental results showed that the proposed scheme can extract keyframes in videos with gradual transition between shots. However, this work can extract similar frames affected by the presence of flashlights.

Salehin and Paul, (2017) presented a keyframe extraction scheme based on human eye movement. The object movements in a given video are determined using the variations in RGB channel in the foveal region around the gaze point of the human retina. If the difference in the color channel is greater than or equal to a threshold value, then an object movement is detected. The distance between gaze points (images) were then computed after determining smooth pursuit. If the distance value is zero, then there is no object movement. Lastly, images are arranged downward according to the distance. The images at the top of the order are then selected as keyframes. Although, the approach detects camera movements and variation in illumination present in a video, it failed in the detection of any form of the gradual transitions leading to the extraction of redundant frames. In addition, the approach is not reliable as it is based on perception of the human eye.

Satpute and Khandarkar, (2017) presented a correlation approach for selecting representative frames. The frames from the original video were first broken into individual frames. The similarities/dissimilarities between two consecutive frames was then determined by computing the correlation for each of their color channels, and simultaneously comparing between the frames. The duplicated frames were deleted, and the unique frames extracted were saved as representative frames. The authors also employed a parallel processing task to reduce the computational time of the proposed technique. Although the proposed technique can extract representative images at a very low computational time. However, this work failed to detect transitions between shots. Hence, extracting redundant key frames.

Wu *et al.*, (2017) presented a novel approach for the extraction of keyframes using High Density Peaks Search Clustering (HDPSC). The proposed technique was carried out in three major steps. In the first step, an SVD approach was utilized to extract the video frames and eliminate the duplicated ones from the given video. In the second step, a Bag of Word (BoW) model was utilized for the extraction of local features on the extracted frames, and represent them with their corresponding histograms. Finally, the HDPSC approach was utilized to cluster the frames, and from each cluster the frame that is at the centroid is selected as a keyframe. Although, the proposed approach provides a more condensed version of the original video. However, some keyframes were missed during the extraction process due to their similarities in local features with other frames in the video.

Lv and Huang, (2018) implemented an improved Nearest Neighbor Clustering Approach (NNCA) for the selection of representative frames. A motion blur detector was also employed to detect unclear images in the video frames. The blurry frames detected were filtered out using Gaussian filter and Laplace operator. The NNCA directly partitioned the video frames into a number of clusters instead of segmenting them into shots. The smaller clusters were then integrated into the closest larger cluster. Finally, the frames nearest to the centroid of the larger clusters were extracted as the key frames. The experimental result demonstrates that the improved NNCA has low computational time due to the shot segmentation skipped. Also, all the representative frames extracted are of high visual quality, and no blurry key frame extracted. However, a number of frames involved in shot transitions can be extracted as key frames due to the lack of utilizing shot boundary detector in the process of extracting the representative frames. Hence, generating more redundant frames.

Rodriguez *et al.*, (2018) presented an approach for the extraction of representative frames based on calculating the absolute difference of histogram. The algorithm was implemented in four modules. Firstly, the images from the original video were segmented into number of shots. Secondly, the histogram and histogram equalization of the individual frames were computed. Thirdly, the absolute difference between successive images was then computed. Lastly, a threshold value was determined by computing the mean and standard deviation of the absolute difference of the equalized histogram. A comparison was then established with the calculated threshold. If the absolute difference between two successive images is higher than its threshold, then the first image is extracted as a representative image. The experimental results showed that the proposed technique is computationally simple and can efficiently extract set of key frames from a video with abrupt shots. However, this work failed in the detection of gradual transitions present in the original video leading to the extraction of redundant frames.

In view of these limitations identified from the review of related works, it is evident that the existing techniques failed to detect gradual transitions and sudden illuminance present in a video shot resulting in the extraction of redundant key frames. To extract a set of unique keyframes, there is need to cluster feature related keyframes that exist in video frames into a single cluster using histogram difference and k-means clustering approach. This significantly improve the compression ratio by eliminating the redundant keyframes so as to achieve a better transmission rate.

CHAPTER THREE

MATERIALS AND METHOD

3.1 Introduction

In this chapter, the materials, methods, and step by step procedures used for the development of the improved keyframe extraction scheme for video summarization based on histogram difference and k-means clustering are discussed. The methodology adopted in carrying out the research is also presented.

3.2 Materials

The materials utilized for the implementation of this research include the following:

i. A computer system with 4G RAM and 2.54GHz processor was used as the processing system.

ii. Videos downloaded from the popular video-sharing website YouTube.

iii. MATLAB/Simulink R2018a.

3.3 Methodology

The methodology adopted in carrying out the research are itemized as follows:

i. Implementation of a shot boundary detection scheme based on histogram difference between consecutive video frames.

   a) Input and read the video file

   b) Generate the total frames in the video

   c) Compute the current status of the number of video shots to zero

   d) Set cycle from the first frame to the last one

   e) Select the current frame

f) Select the next frame

g) Generate the histogram of the current and next frame

h) Compute the histogram difference between the current and next frame

i) Compute a threshold value

j) Establish a condition to detect the transitioned frames

ii. Development of a keyframe extraction scheme based on k-means clustering.

a) Compute the total number of shots obtained in (i)

b) Initialize cycle from the current shot to last shot

c) Select candidate shot set

d) Compute the current status of a number of shot cluster to zero

e) Initialize k centroids

f) Generate frames to the closest centroid

g) Select the frame closest to the centroids as keyframes

iii. Evaluation and Comparison of results of the developed scheme with results obtained from the scheme developed by Rodriguez *et al.* (2018) and Sheena and Narayanan (2015) based on compression ratio, precision and extraction rates, and f-measure.

a) Replication of the work of Rodriguez *et al.* (2018) and Sheena and Narayanan (2015)

b) Comparison of results of the developed scheme with results obtained from the scheme developed by Rodriguez *et al.* (2018) and Sheena and Narayanan (2015)

3.4 Implementation of Histogram Difference Based Shot Boundary Detection Scheme

The step-by-step procedures involved in the implementation of the shot binary detection scheme is discussed in the following subsections. The video shots are detected using a histogram difference-based approach. This approach is carried out in three modules. In the first module, the total frames from the original video file are extracted and stored in a defined location. In the second module, two successive frames are taken and the histogram difference between them is computed. In the final module, a shot boundary is detected by finding transitions between consecutive frames. The pseudocode for the histogram-based approach is given in algorithm 1.

Algorithm 1: Shot Boundary Detection Scheme

```
Input: a new video
Output: E (set of video shots detected)
Step1: Read the video file
Step2: Count number of video frames
Step3: Total frames ← MOV. NumberOfFrame
Step4:  For k = 1 to Total frames
Step5: I ← k
Step6: J ← k+1
Step7: S ←   FrameDiff (I, J)
//end for
Step8: Mean ← mean2(S)
Step9: Standard Deviation ← std2 (S)
Step10: Find the threshold
Step11: threshold ← Standard Deviation + (Mean * a) // a is constant
Step12: if (S >threshold)
Step13: E ← J as the beginning of a new shot
//end if
```

3.4.1 Generation of Video Frames

The first stage of the keyframe extraction system is video acquisition. The videos used for this research were downloaded from the popular video-sharing website YouTube. These videos are of different sizes and contents, and can be assessed through this link:

‘https://drive.google.com/drive/u/0/folders/1TpssNoBwtczuQvco86yfd4EGf9c8ZmiQ’.

The obtained videos are grouped into 4 categories namely; surveillance footage, movie clip, advert, and sport. A sample of these videos is shown in Figure 3.1. These videos are either low or high motion videos, and the entire operations of the system are performed on them. The surveillance footage is a low motion due to its static background, while the remaining three videos (i.e., movie clip, advert, and sport) are high motion videos due to the rapid change in background and actions. Table 3.1 describes the videos used.

Table 3.1: Videos Description

S/n	**Name**	**Type**	**Format**	**Duration(sec)**	**Size(mb)**
1	Advert	High motion	AVI	29	3.87
2	Surveillance	Low motion	MP4	7	6.57
3	Movie clip	High motion	MP4	8	3.59
4	Sport	High motion	MP4	5	0.443

Figure 3.1: Sample of Video Dataset

Figure 3.1 depicts samples of videos used. These videos are made up of a number of frames that were extracted and stored in a defined location using the program code shown in Figure 3.2.

```
Filename = 'Advert.avi';
MOV = VideoReader(filename);
opFolder1 = fullfile('C:\MATLAB\Keyframe\Totalframes');
NumberOfFrame = MOV.NumberOfFrames
```

Figure 3.2: Snippet Code of Generating Video Frames

Figure 3.2 depicts a snippet code for the extraction and storage of the video frames. The total number of frames in each video varies due to the variation in visual contents and resolutions.

Figure 3.3 shows the sample of frames extracted from the advert video which consists of a total number of 746 frames. These extracted images are what constitute the entire activities of the advert video, and most of the frames are redundant as they are repetition of similar activities.

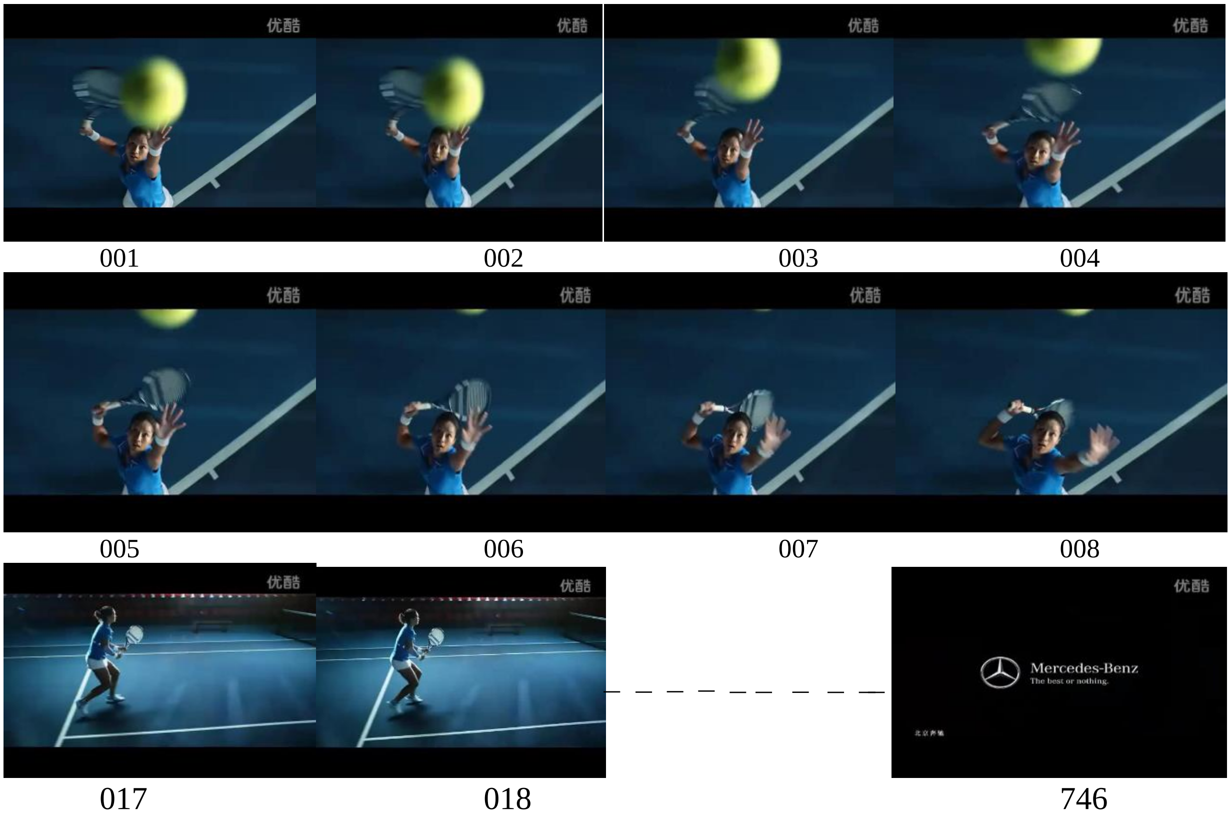


Figure 3.3: Sample of Advert Frames

Figure 3.4 shows the sample of frames extracted from the sport video which consist of a total number of 173 frames. These extracted frames are the still representation of the entire movement in the sport video. As seen from Figure 3.4, series of neighboring frames depicts similar activity and as such resulting in more redundant frames.

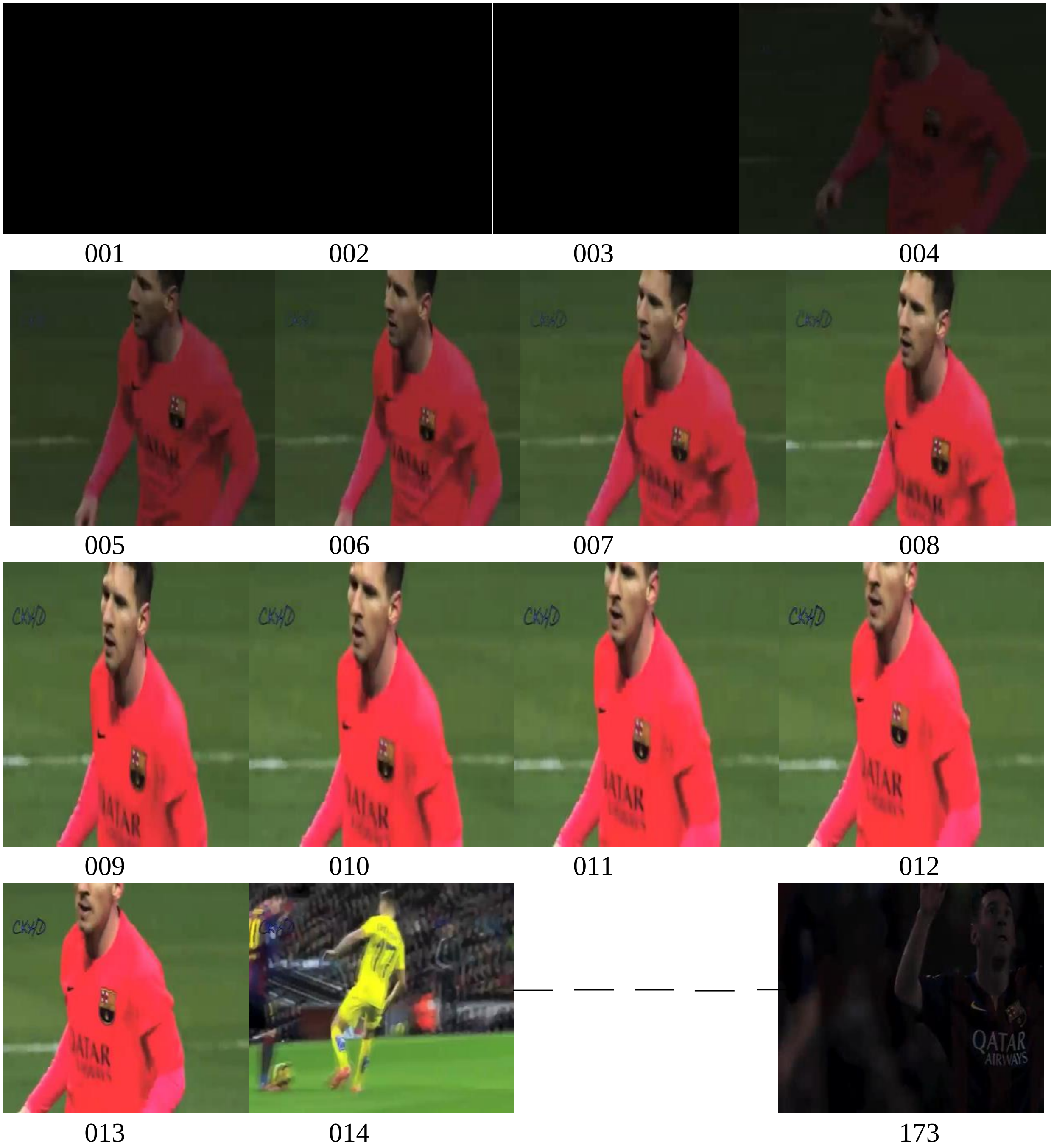
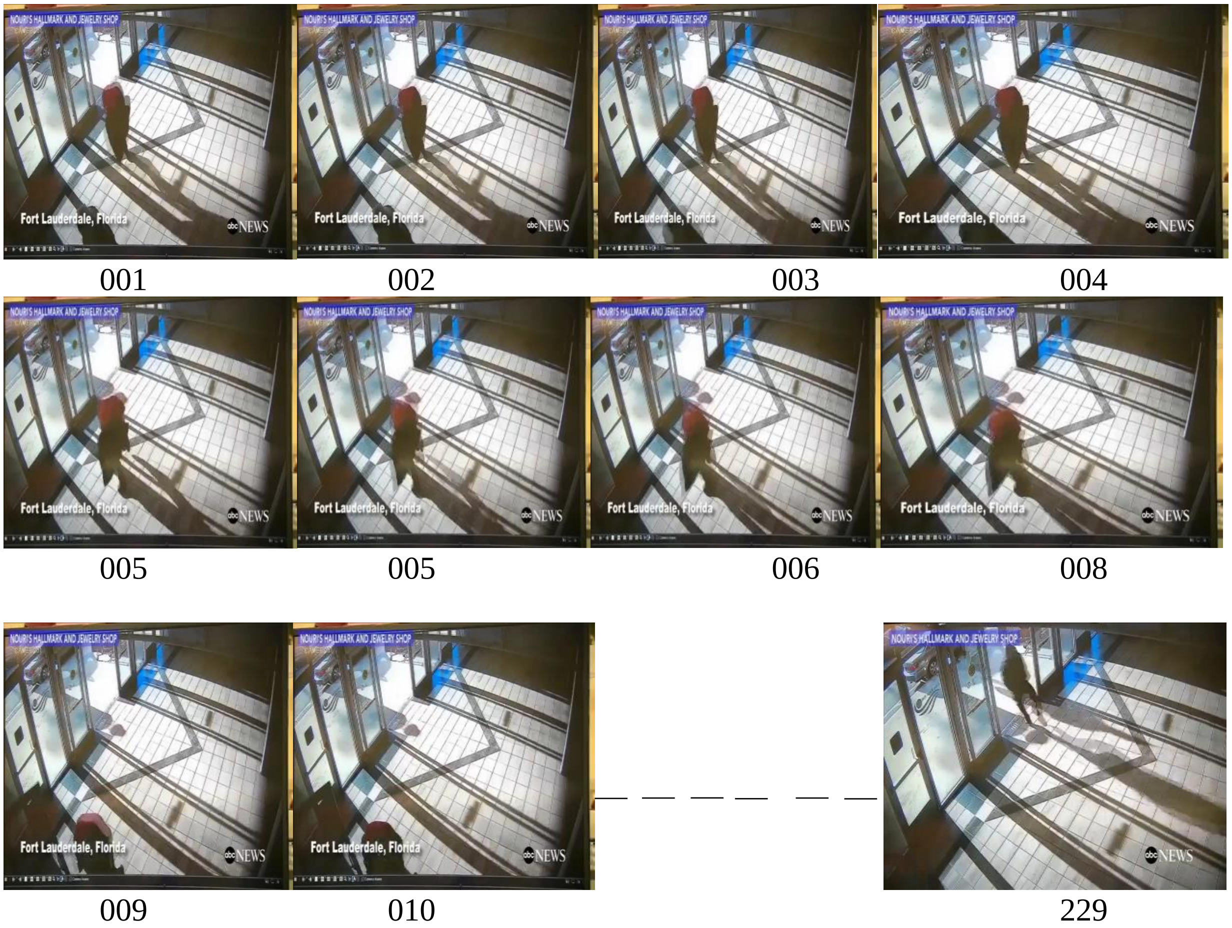


Figure 3.4: Sample of Sport Frames

Figure 3.5 shows the sample of frames extracted from the surveillance video. This video consists of a total number of 229 frames, most of which represent similar visual contents. As seen from Figure 3.5, frame 001 to frame 012 depicts similar foreground activities. The changes between these extracted frames only occur at the foreground, and this is as a result of the surveillance footage being a low motion video.



001 002 003 004

005 005 006 008

009 010 229

Figure 3.5: Sample of Surveillance Frames

Figure 3.6 shows the sample of frames extracted from the movie clip which consists of a total number of 254 frames. Most of the extracted frames are affected by video editing effects such as fade-in/out. These editing effects were added during the course of production to make the video suitable and entertain for the user. In the process of viewing this video, the gradually transitioned frames (i.e., frame 001 to frame 007) are not visible to the human eye but become visible when each video frame is extracted as shown in Figure 3.6. These transitioned frames increase the number of redundant frames in the video.

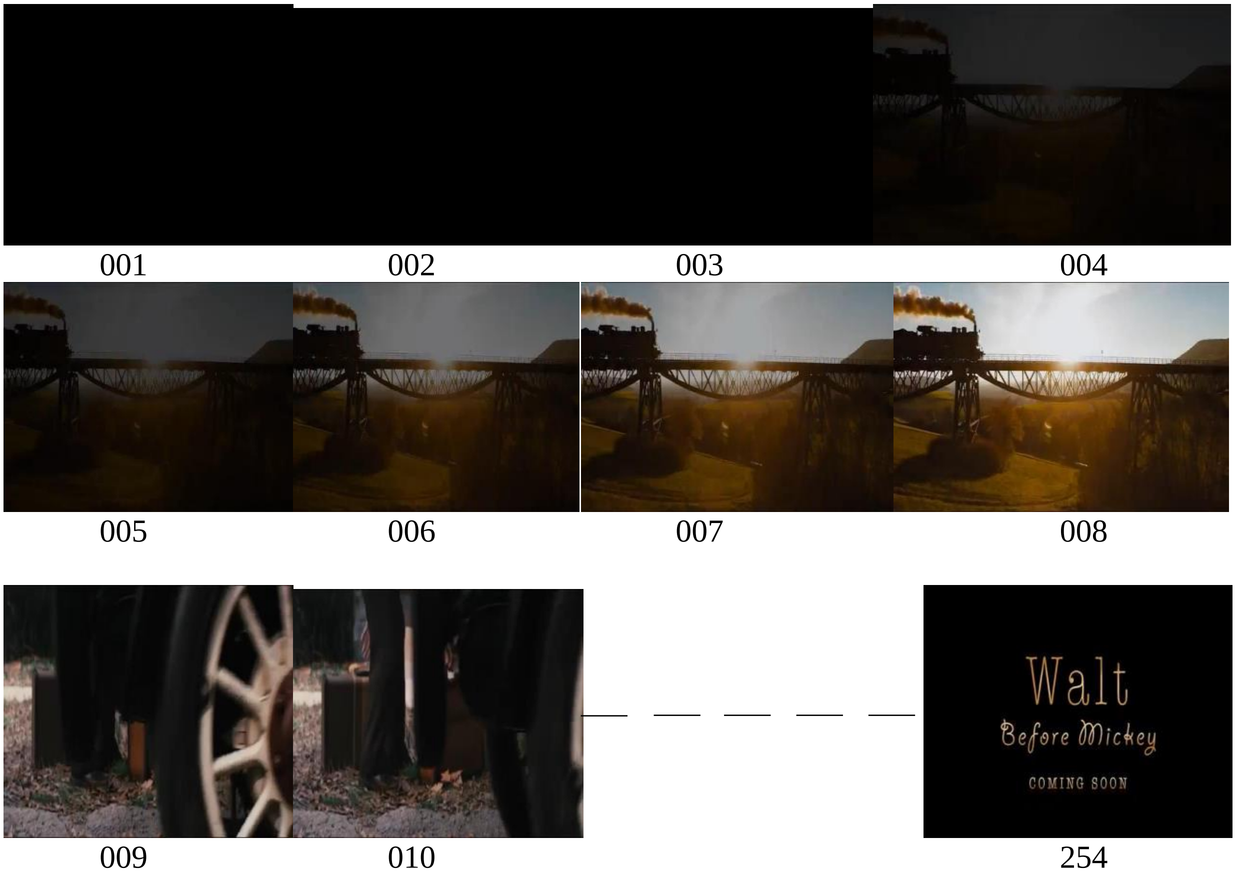


Figure 3.6: Sample of Movie Clip Frames

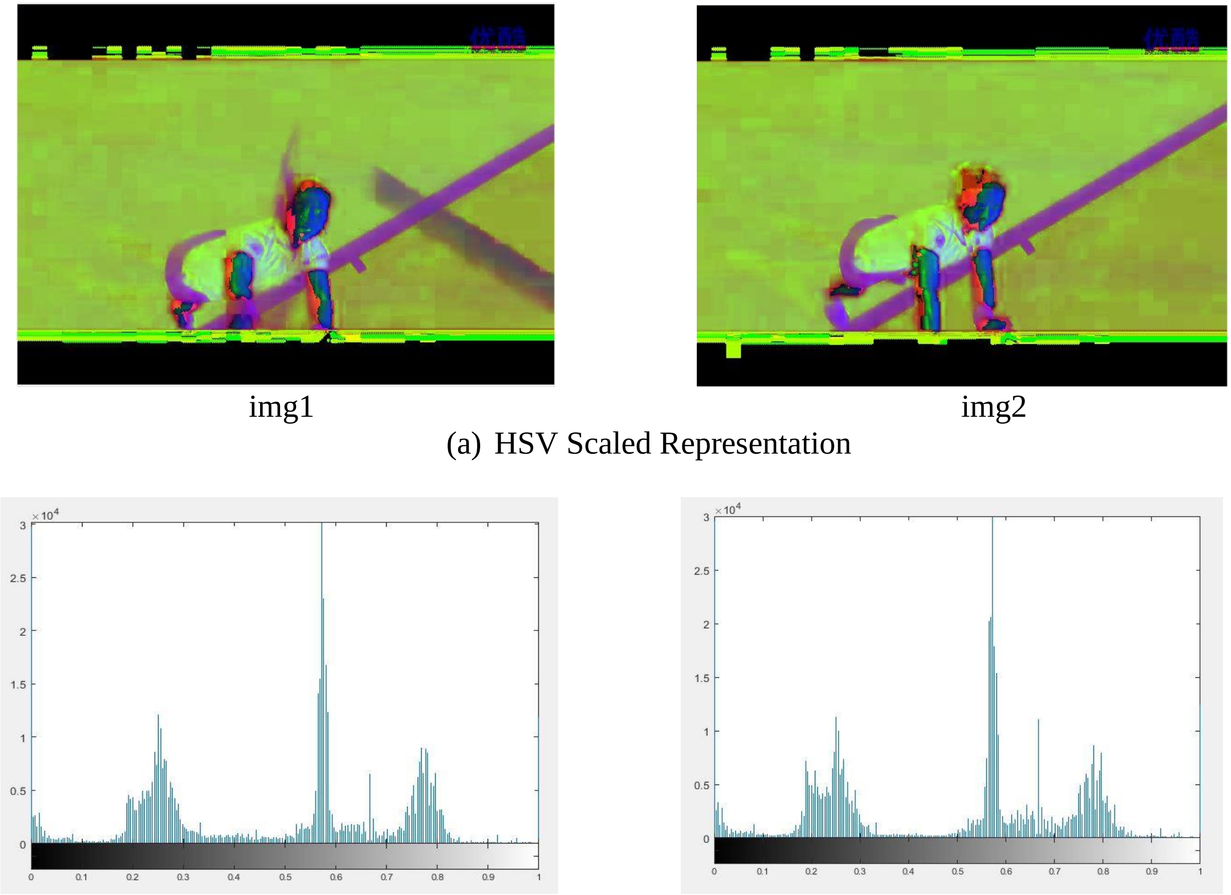
3.4.2 Computing Histogram Difference Using Equation (2.2)

In this subsection, the histograms of successive frames are first taken to compute the differences between the frames. A function is then created to convert the video frames into their corresponding HSV scaled images for easy computation. Using the first and second frames in Figure 3.3, the program code used for converting the frames to HSV and generating their respective histograms is shown in Figure 3.7.

```
hsv1=rgb2gray(im1);
hsv2=rgb2gray(im2);
f11=imhist(k);
f12=imhist(l);
```

Figure 3.7: Snippet Code for Generating Frames Histogram

The output result of Figure 3.7 is the histograms and HSV scale of the two frames as shown in Figure 3.8.

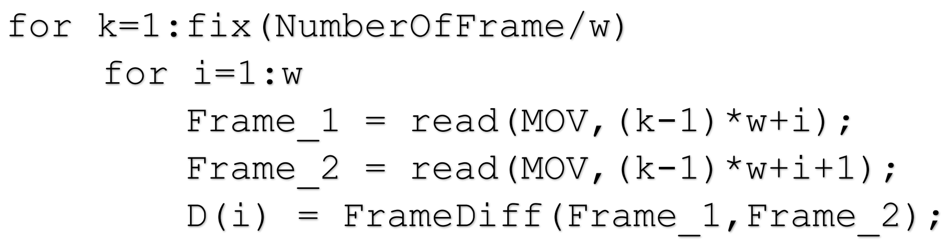


(a) HSV Scaled Representation

(b) Histogram Representation

Figure 3.8: Frames Representation

For every iteration, successive HSV scaled frames are taken and their histogram the difference is computed using equation (2.2). The sum of the elements of the histogram is then computed and returned. Figure 3.9 shows the program code for calculating the difference between successive frames.

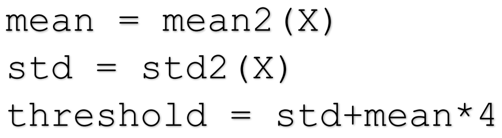


```
for k=1:fix(NumberOfFrame/w)
    for i=1:w
        Frame_1 = read(MOV,(k-1)*w+i);
        Frame_2 = read(MOV,(k-1)*w+i+1);
        D(i) = FrameDiff(Frame_1,Frame_2);
```

Figure 3.9: Snippet Code of Computing Frames Difference

3.4.3 Shot Boundary Detection

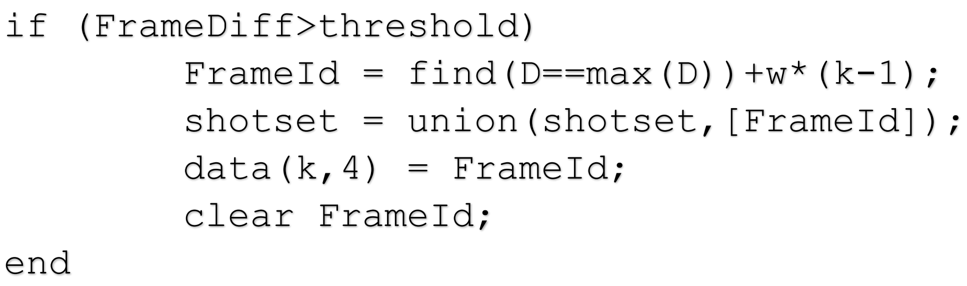
In this subsection, the visual changes between two consecutive video shots are computed through the establishment of a condition between a threshold value and the frame difference. The threshold value is utilized to identify the frontier between shots, and it is determined by computing the mean and standard deviation of the histogram differences obtained using equations (2.3), (2.4), and (2.5) respectively. Figure 3.10 shows a snippet of the program code used for calculating the threshold value.



```
mean = mean2(X)
std = std2(X)
threshold = std+mean*4
```

Figure 3.10: Snippet Code of Computing Threshold Value

Now, for every iteration, the calculated threshold value is compared with the frames difference computed previously using expression 2.2. If the frames difference for the two consecutive frames is greater than the threshold value, then the second frame of that pair is selected and taken as the beginning of a new video shot, otherwise, the first frame of the pair is considered. Finally, after executing each iteration, sets of video shots are obtained and stored in a defined directory. Figure 3.11 depicts the snippet code for detecting a shot boundary.



```
if (FrameDiff>threshold)
        FrameId = find(D==max(D))+w*(k-1);
        shotset = union(shotset,[FrameId]);
        data(k,4) = FrameId;
        clear FrameId;
end
```

Figure 3.11: Snippet Code for Detecting Shot Boundary

3.5 Development of K-Means Clustering Based Keyframe Extraction Scheme

The processes involved in the development of the keyframe extraction scheme are discussed in detail in the following subsections. The keyframe extraction scheme was developed using the k-means clustering approach. This section consists of two subsections. In the first subsection, the clusters within video shots are determined and the k centroid is computed. In the second subsection, the frames closest to the centroid from the respective clusters are extracted as representative. Pseudocode for K-Means is given in algorithm 2.

Algorithm 2: Developed Scheme

```
Input: E (video shots)
Output: Key (keyframes)
Step1: Read the video shot
Step2: find initial centroid
Step3: Initial centroid, Cj ← mean(c1, c2,….ck)
Step4:  For Di ← (1<= i <=n)
Step5: Find the closest frame
Step6: Key ← (Di, Cj)
//end for
Step7: Repeat
Step8: For new cluster, Di ← (i <= Cj)
Step9: Frame stays in the cluster, Di ← i
//else
Step10: New cluster is form
//end for
Step 12: Return assignment
```

3.5.1 Generation of Shot Clusters

In this subsection, the frames within a candidate shot are clustered based on the variations between their features. These features are the visual characteristics of the frames and are computed using the snippet code shown in Figure 3.12. It is possible to have more than one cluster within a video shot.

```
function FeatureOfImg = FeatureOfImg(img)
K_define;
gray = rgb2gray(img);
[row,col] = size(gray);
weight = fix(row/K);
height = fix(col/K);
FeatureOfImg = [];
for i=1:K
   for j=1:K
temp=imcrop(gray,[((i-1)*weight),((j-1)*height),weight,height]);
       m = mean2(temp);
       var = std2(temp);
       FeatureOfImg = [FeatureOfImg, m var];
       clear m;
       clear var;
    end
end
```

Figure 3.12: Snippet Code for Frame Feature Extraction

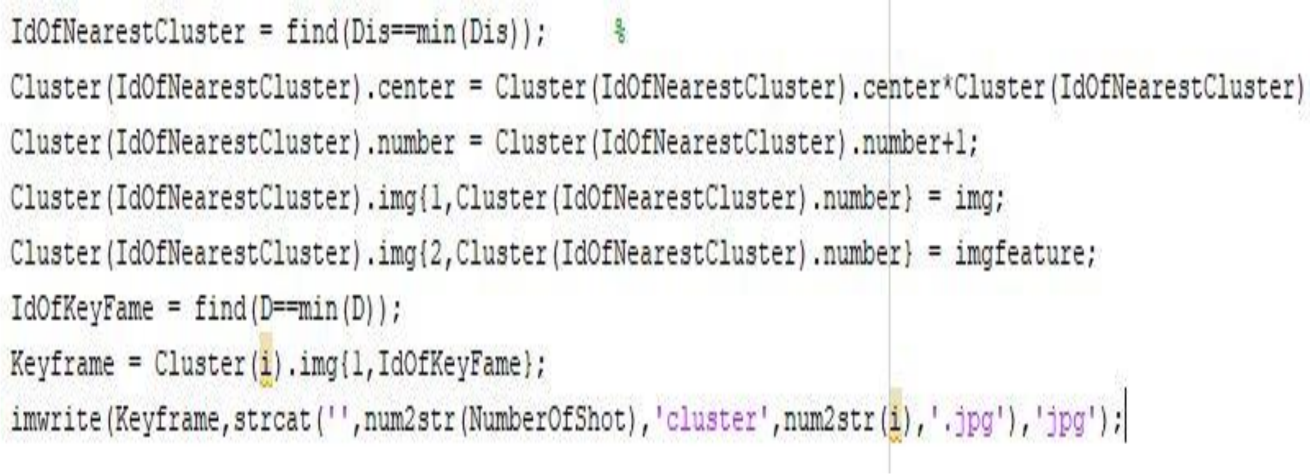
Based on the variations between the features of the frames, a centroid is computed using the program code shown in Figure 3.13. Each cluster in the video shot is determined by its member frames and centroid. The centroid for each cluster is the point at which the sum of distances from all the frames in that cluster is minimized.

```
for k=1:(NumberOfShot-1)
    F = read(MOV,shotset(k)+1);
    NumberOfCluster = 1;
    Cluster(1).center = FeatureOfImg(F);
    Cluster(1).number = 1;
    Cluster(1).img = cell(2,1);     %
    Cluster(1).img{1,1} = F;
    Cluster(1).img{2,1} = FeatureOfImg(F);
```

Figure 3.13: Snippet Code for Computing Cluster Centroid

3.5.2 Keyframe Extraction

In this subsection, the variation between cluster centers and the frames within them was computed and the frame with minimum distance to the centroid was extracted as a keyframe. Figure 3.14 shows a snippet code for extracting keyframes.



```
IdOfNearestCluster = find(Dis==min(Dis));     %
Cluster(IdOfNearestCluster).center = Cluster(IdOfNearestCluster).center*Cluster(IdOfNearestCluster)
Cluster(IdOfNearestCluster).number = Cluster(IdOfNearestCluster).number+1;
Cluster(IdOfNearestCluster).img{1,Cluster(IdOfNearestCluster).number} = img;
Cluster(IdOfNearestCluster).img{2,Cluster(IdOfNearestCluster).number} = imgfeature;
IdOfKeyFame = find(D==min(D));
Keyframe = Cluster(i).img{1,IdOfKeyFame};
imwrite(Keyframe,strcat('',num2str(NumberOfShot),'cluster',num2str(i),'.jpg'),'jpg');
```

Figure 3.14: Snippet Code of Extracting Keyframes

Figure 3.14 depicts a snippet code for extracting representative frames in the video shots. The total number of clusters in every video shot varies, and as such more than one keyframe may be extracted from a single shot.

3.6 Comparing the Performance of the Developed Scheme with the Existing Scheme

To evaluate the performance of the developed scheme certain metrics are used. These metrics include compression ratio, precision and extraction rates, and f-measure. They are explained in section 2.2.7. The four different videos trained with the developed scheme were used to evaluate its performance as well as compared with the existing schemes.

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the results obtained from the research are presented and discussed. The performance of the existing techniques and the developed scheme are evaluated using the performance metrics as discussed in subsection 4.3

4.2 Simulation Results

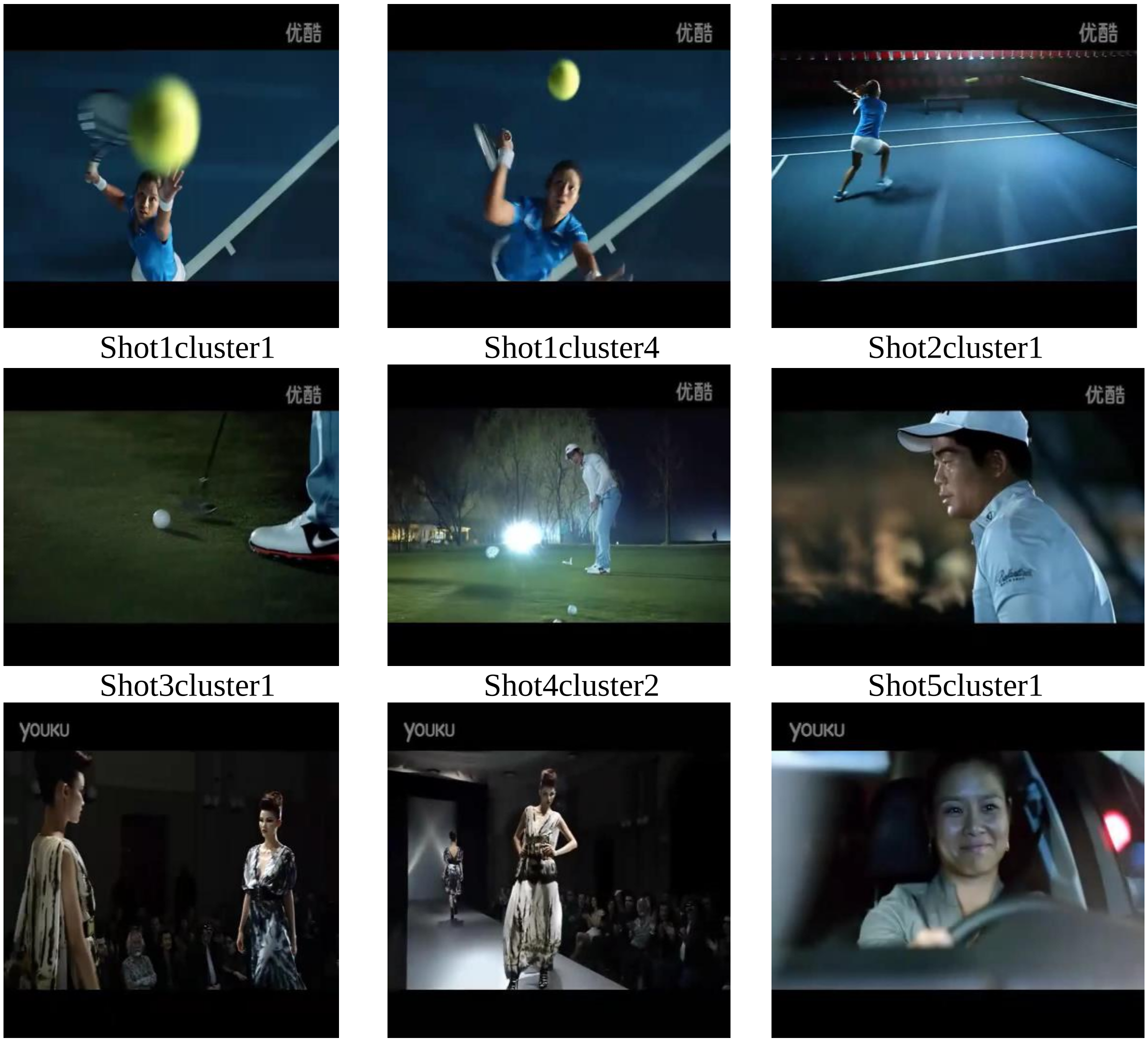
In this section, the keyframes extracted from the videos used are presented. The acquired videos used illustrated different challenges such as camera motion, presence of flashlights, and gradual transitions. Table 4.1 shows the total video frames and the keyframes extracted from each of the videos used.

Table 4.1: Simulation Results

Videos used	**Total Frames**	**Video Shots**	**Keyframes**		
			Developed Scheme	**Rodriguez *et al.*, (2018)**	**Sheena and Narayanan (2015)**
Advert	746	27	47	173	181
Surveillance	229	11	11	101	114
Movie clip	254	13	18	40	62
Sport	173	8	23	35	53

Table 4.1 depicts the total number of frames in the original videos, and their corresponding keyframes extracted by both the developed and existing schemes. It can be seen that the existing schemes extracted a higher number of representative frames compared to the developed scheme. This is as a result of the extraction of feature related and gradual transitioned frames. However,

the developed scheme was able to reduce these redundant frames by clustering the similar frames and extracting the most representative one among them as a keyframe without affecting the integrity of the frames. Figures 4.1 to 4.4 shows the simulation result of the developed scheme when tested on each of the videos used. The results were compared with that of Rodriguez *et al.*, (2018) and Sheena and Narayanan (2015) using the compression ratio, precision and extraction rates, and f-measure.



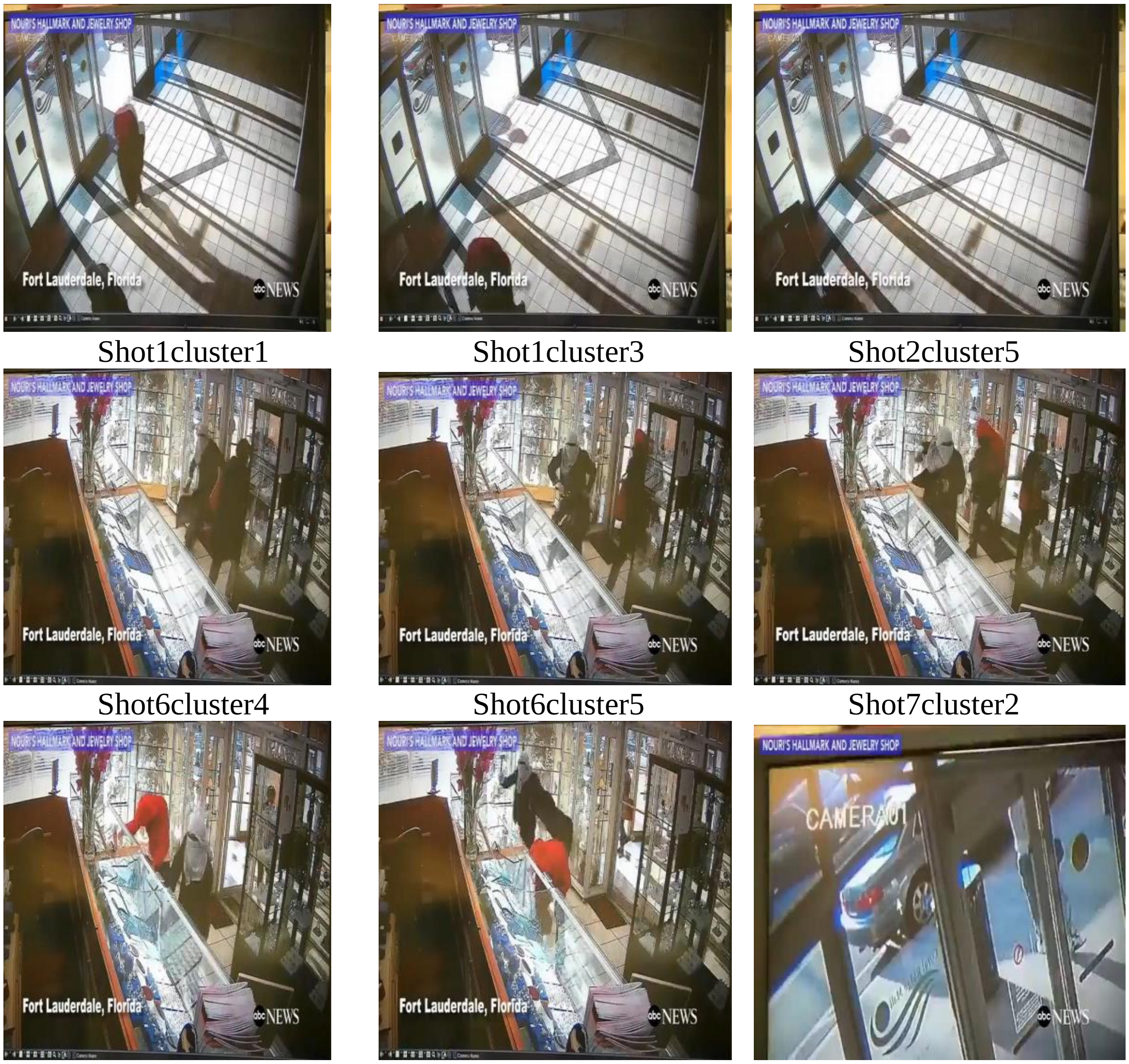
Shot1cluster1 Shot1cluster4 Shot2cluster1

Shot3cluster1 Shot4cluster2 Shot5cluster1

Shot7cluster1 Shot11cluster2 Shot12cluster2

Figure 4.1: Sample of Advert Keyframes

Figure 4.1 shows some keyframes extracted from the advert video by the developed scheme. These keyframes represent the entire visual contents of the original video, and no two or more feature related frames were extracted.



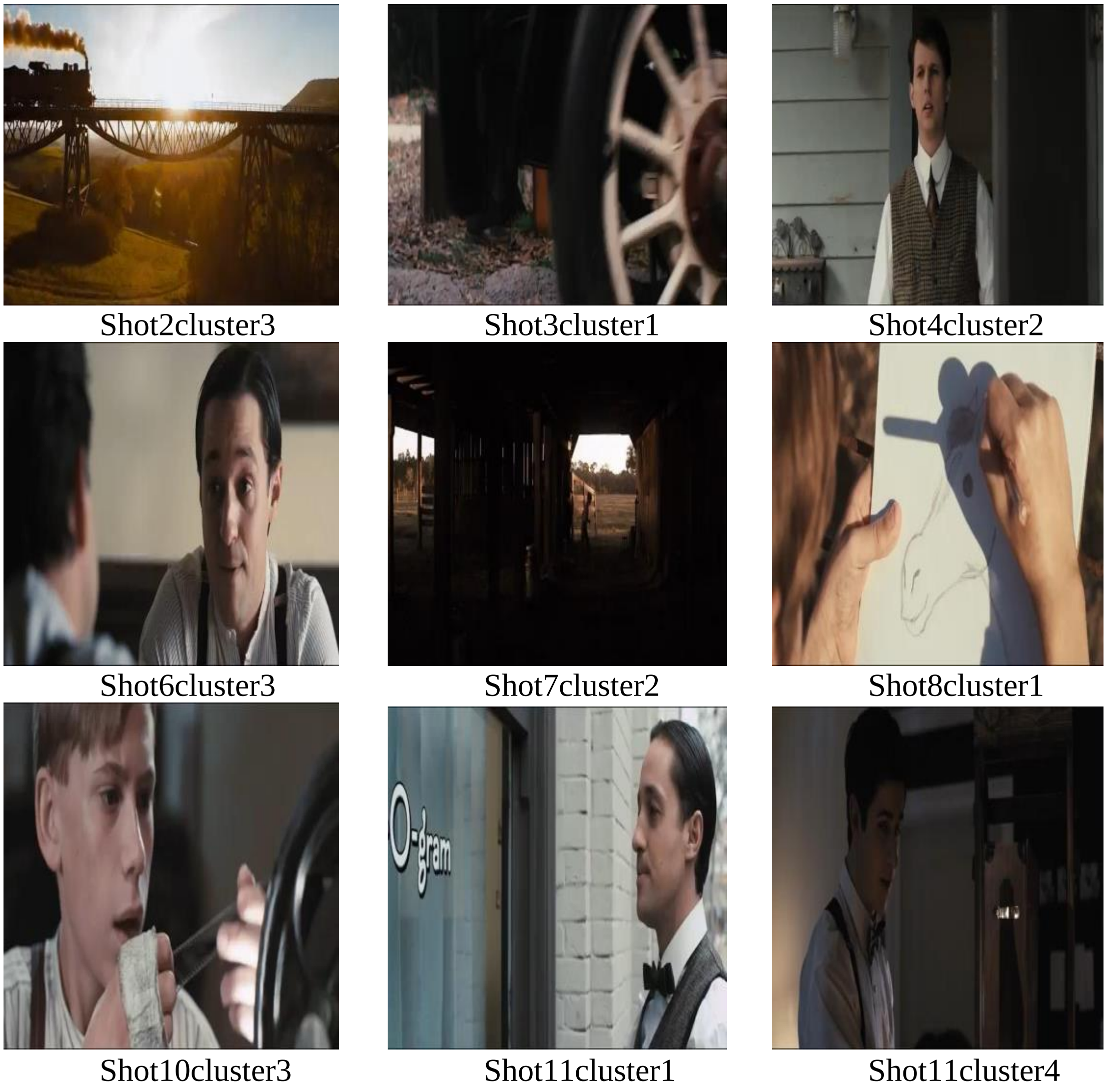
Shot1cluster1 Shot1cluster3 Shot2cluster5

Shot6cluster4 Shot6cluster5 Shot7cluster2

Shot8cluster3 Shot10cluster3 Shot11cluster1

Figure 4.2: Sample of Surveillance Keyframes

Figure 4.2 shows sample of keyframes extracted from the surveillance video. Mostly, surveillance video frames are not affected by any gradual transitions as it is captured in real time. Therefore, the developed scheme extracted only set of unique frames that depict the key events of the entire surveillance footage.



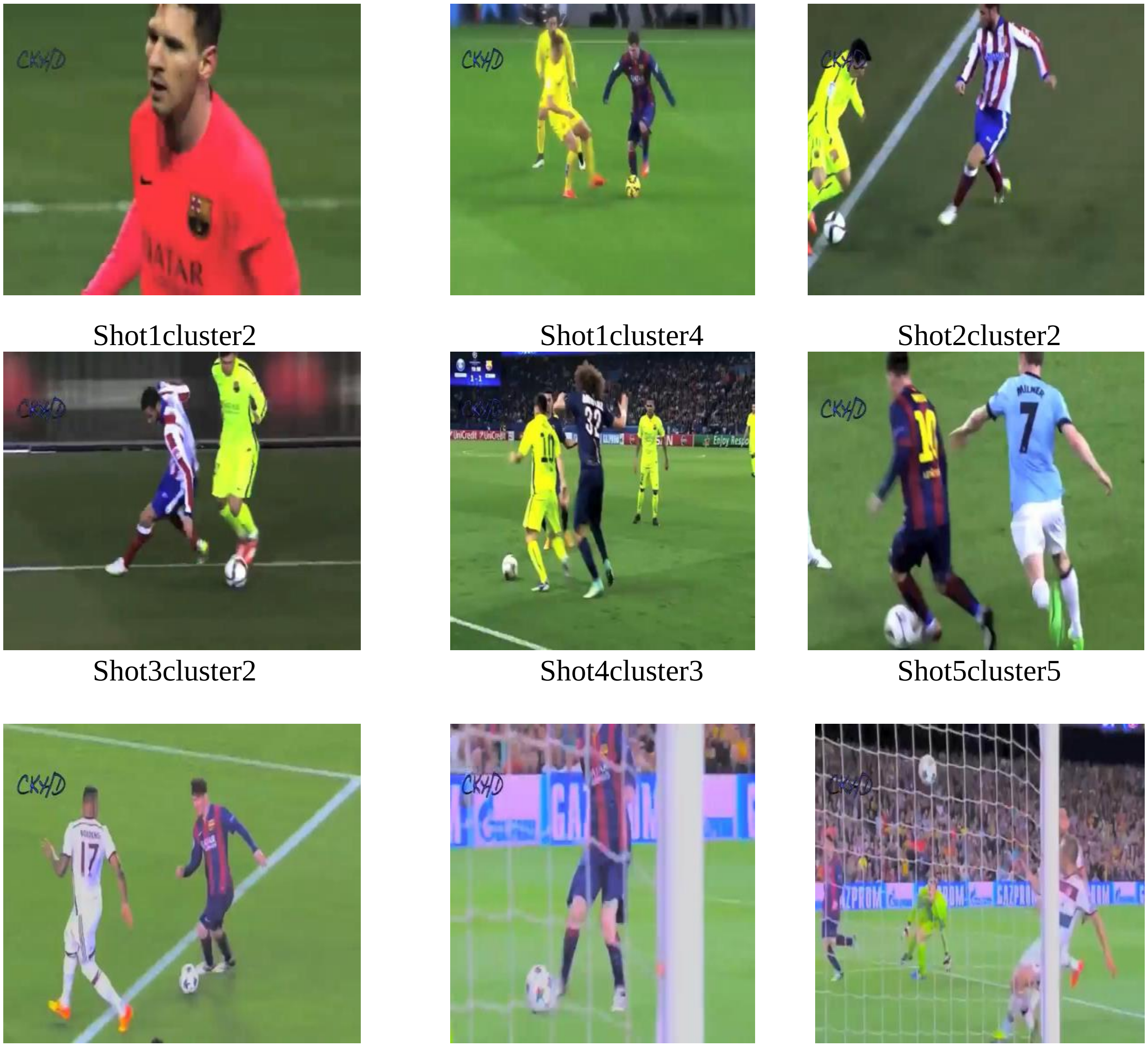
Shot2cluster3 Shot3cluster1 Shot4cluster2

Shot6cluster3 Shot7cluster2 Shot8cluster1

Shot10cluster3 Shot11cluster1 Shot11cluster4

Figure 4.3: Sample of Movie Clip Keyframes

Figure 4.3 shows the sample of keyframes extracted from the movie clip by the developed scheme. During the course of production, movies undergo lots of video editing processes resulting in more redundant frames. However, as seen in Figure 4.3, frames affected by these video editing effects were eliminated. As a result, only unique set of keyframes were extracted.



Shot1cluster2 Shot1cluster4 Shot2cluster2

Shot3cluster2 Shot4cluster3 Shot5cluster5

Shot7cluster5 Shot8cluster1 Shot8cluster4

Figure 4.4: Sample of Sport Keyframes

Figure 4.4 depicts a sample of keyframes extracted to represent an entire football match. These keyframes provide a highlight of the main activities from the beginning to the end of the match.

4.3 Performance of the Developed Scheme on the Acquired Videos

To evaluate the performance of our developed keyframe extraction method, the compression ratio, precision and extraction rates, and f-measure were used as metrics.

4.3.1 Evaluation Using Compression Ratio

The compression ratio was evaluated using equation (2.6). The result obtained from each keyframe as depicted in Figure 4.1 to 4.4 is shown in Table 4.2.

Table 4.2: Compression Ratio of Developed and Existing Schemes

Videos	Developed Scheme	Rodriguez *et al.* (2018)	Sheena and Narayanan (2015)
Advert	93.70%	76.81%	75.74%
Surveillance	95.20%	55.90%	50.40%
Movie clip	92.91%	84.25%	75.59%
Sport	86.71%	79.77%	69.94%

Table 4.2 presents the comparison of the compression ratio of the keyframes depicted in Figures 4.1 to 4.4. It can be seen that video summarization using the developed scheme provides a more condensed version of the full-length videos compared with the existing techniques. These summarized videos provided by the developed scheme are of high quality as there is no any form of degradation in the videos during the extraction process. Also, it can be observed from Table 4.2 that the existing techniques provide a lower compression ratio compared to the developed scheme. This is due to the extraction of multiple feature related frames as keyframes by the existing schemes.

4.3.2 Evaluation Using Precision and Extraction Rates

To determine the precision and extraction rates of the developed and existing schemes, equations (2.7) and (2.8) were utilized. Tables 4.3, 4.4, and 4.5 show the precision and extraction rates of the developed and existing schemes respectively.

Tables 4.3: Precision and Extraction Rates using the Developed Scheme on the Videos

Videos	N_a	N_k	N_m	Precision (%)	Extraction Rates (%)
Advert	45	47	0	95.74	100
Surveillance	11	11	0	100	100
Movie clip	16	18	0	88.89	100
Sport	22	23	0	95.65	100

Tables 4.4: Precision and Extraction Rates using the Existing Scheme of Rodriguez *et al.*, (2018)

Videos	N_a	N_k	N_m	Precision (%)	Extraction Rates (%)
Advert	149	173	3	83.13	98.03
Surveillance	98	101	0	97.03	100
Movie clip	28	40	0	70	100
Sport	35	35	0	100	100

Tables 4.5: Precision and Extraction Rates using the Existing Scheme of Sheena and Narayanan (2015)

Videos	N_a	N_k	N_m	Precision (%)	Extraction Rates (%)
Advert	149	173	3	83.13	98.03
Surveillance	96	101	8	96.00	92.31
Movie clip	25	40	5	62.50	83.33
Sport	35	35	0	100	100

Tables 4.6 show the average precision and extraction rates of the developed and existing schemes respectively.

Tables 4.6: Average Precision and Extraction Rates of Developed and Existing Schemes

Techniques	Precision (%)	Extraction Rates (%)
Developed	95.07	100
Rodriguez *et al.*, (2018)	87.54	99.51
Sheena and Narayanan (2015)	85.41	93.42

The results presented in Table 4.6 shows that the developed scheme obtained higher precision and extraction rates when compared with the existing schemes. This can be attributed to the utilization of the shot boundary scheme and k-means clustering approach of extracting keyframes in the developed scheme. Figure 4.5 shows the bar chart of the results obtained by the developed scheme compared to the existing techniques based on precision and extraction rates.

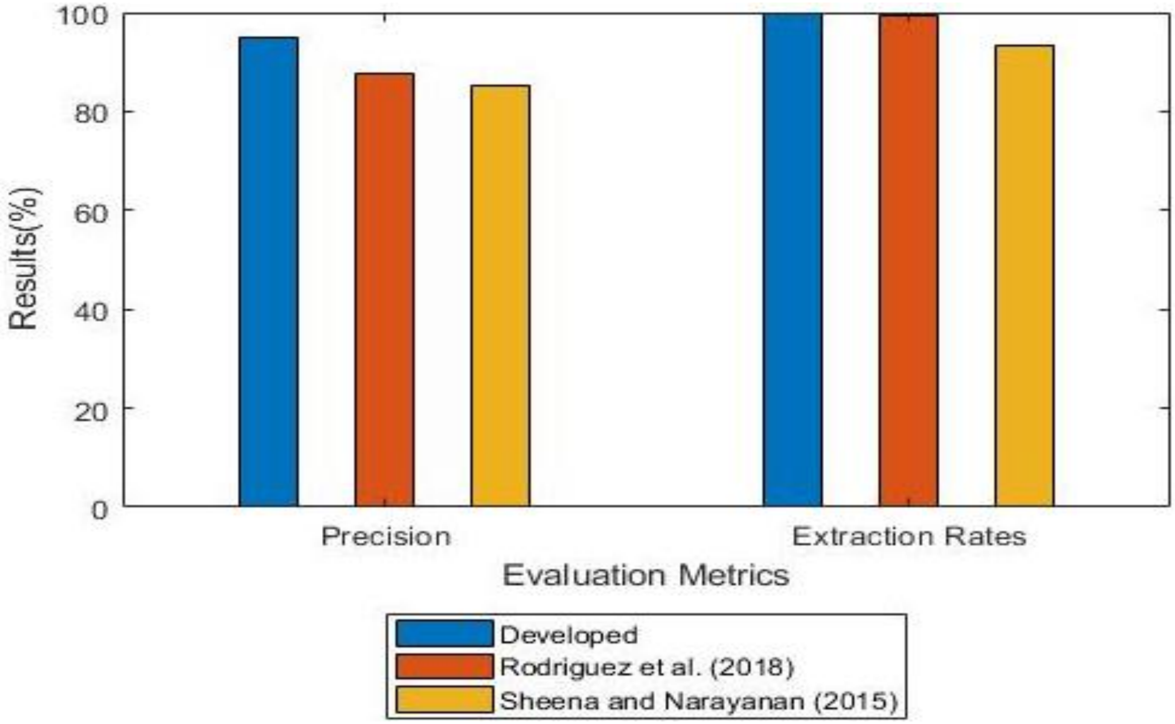


Figure 4.5: Comparison of Results Based on Precision and Extraction Rates

Figure 4.5 depicts the results of the comparison obtained by the developed scheme and existing schemes based on precision and extraction rates. It can be seen the developed scheme outperformed the existing schemes by 8.60% and 11.31% in precision rate, and 0.49% and 7.04% in extraction rates respectively.

4.3.3 Evaluation Using Average F-Measure

To compute the f-measure, the average precision and extraction rates of the schemes presented in Table 4.6 are combined using equation (2.9). Table 4.7 shows the average f-measure of the developed and existing schemes respectively.

Tables 4.7: F-Measure Rates of Developed and Existing Schemes

Techniques	F-Measure (%)
Developed	97.47
Rodriguez *et al.,* (2018)	93.14
Sheena and Narayanan (2015)	89.24

The results presented in Table 4.7 shows that the developed scheme obtained higher f-measure rate when compared with the existing schemes. Figure 4.6 shows the bar chart of the results obtained by the developed scheme compared to the existing techniques based on the f-measure rate.

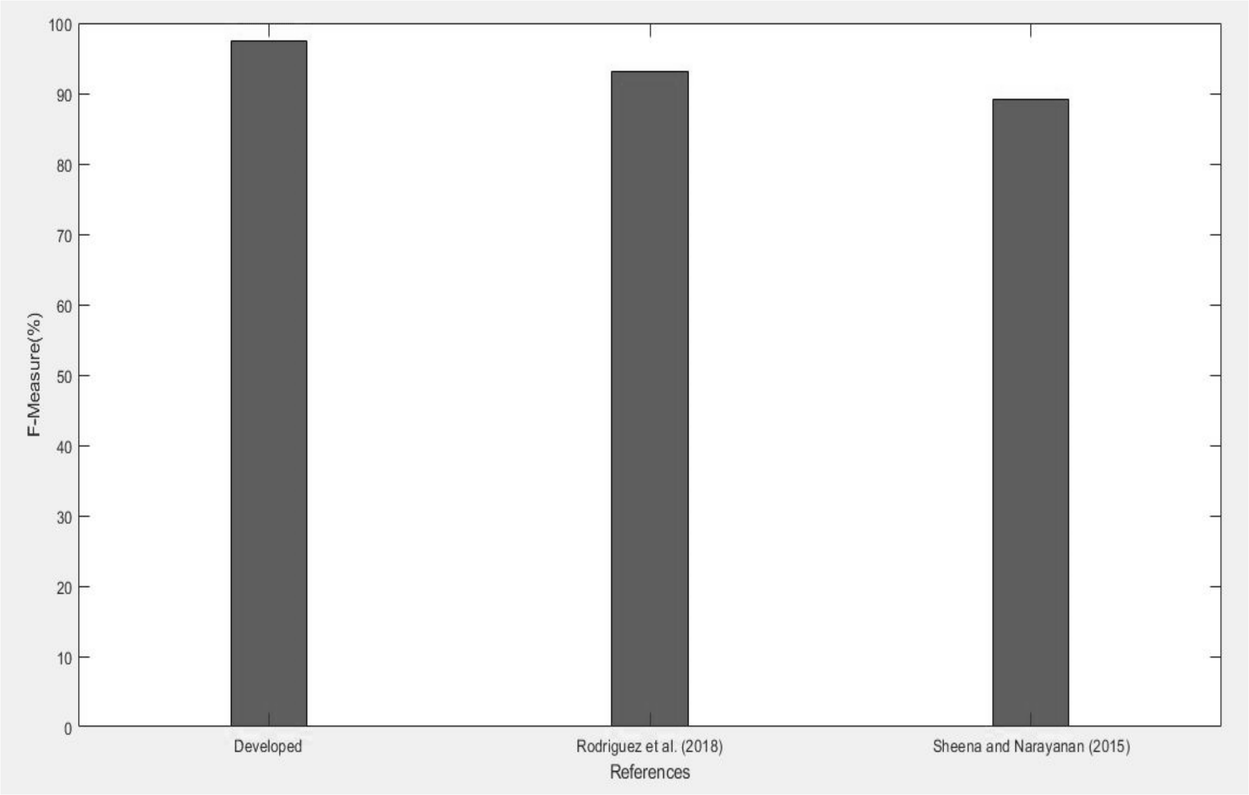


Figure 4.6: Comparison of Results Based on F-Measure Rate

Figure 4.6 depicts the results of the comparison obtained by the developed scheme and existing schemes based on the f-measure rate. It can be seen that the developed scheme outperformed the existing schemes that utilized the histogram-based approach by 4.65% and 9.22%.

4.3.4 Results of Comparison Between the Developed and Existing Schemes

This subsection presents the results obtained using the developed and existing approaches and the comparison between the two approaches. The summary of the results is presented in Tables 4.8 and 4.9.

Table 4.8: Comparison Between the Developed Scheme and Rodriguez *et al.*, (2018)

Performance Metrics	Developed Scheme	Existing Scheme (%)	Improvement (%)
Compression Ratio (Ave. Val)	92.13%	74.18	24.20
Precision (Ave. Val)	95.07	87.54	8.60
Extraction Rate (Ave. Val)	100	99.51	0.49
F-Measure (Ave. Val)	97.47%	93.14	4.65

Table 4.9: Comparison Between the Developed Scheme and Sheena and Narayanan (2015)

Performance Metrics	Developed Scheme	Existing Scheme (%)	Improvement (%)
Compression Ratio (Ave. Val)	92.13%	67.92	35.65
Precision (Ave. Val)	95.07	85.41	11.31
Extraction Rate (Ave. Val)	100	93.42	7.04
F-Measure (Ave. Val)	97.47%	89.24	9.22

Tables 4.8 and 4.9 presents the summary of the results obtained in video summarization using the developed scheme and the existing schemes. Figure 4.7 shows the bar chart of the developed scheme compared to the existing techniques of Rodriguez *et al.*, (2018) and Sheena and Narayanan (2015).

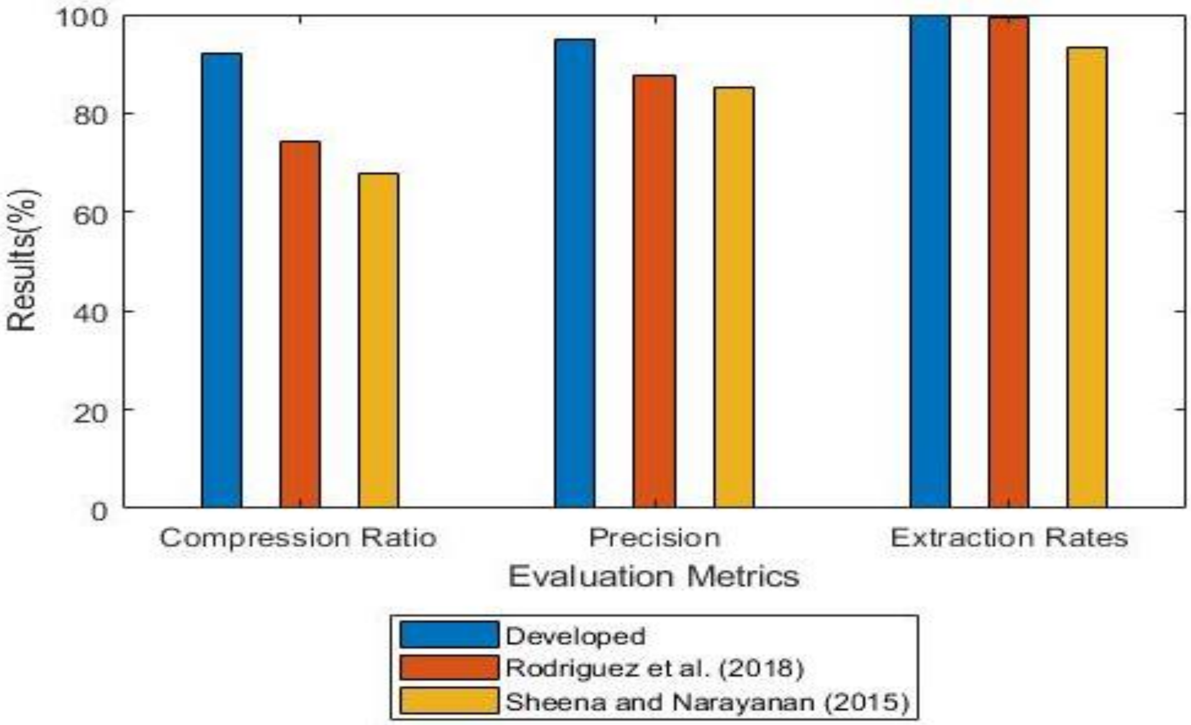


Figure 4.7: Results of Comparison of Developed and Existing Techniques

Figure 4.7 depicts the comparison results of the developed scheme and existing schemes based on compression ratio, precision and extraction rates, and f-measure. It can be seen that the developed scheme outperformed the existing schemes of Rodriguez *et al.*, (2018) and Sheena and Narayanan (2015) by 24.20% and 35.65% in terms of compression ratio. In terms of precision, it outperformed the existing schemes by 8.60% and 11.31%. Also, in terms of extraction rate, it outperformed the existing schemes by 0.49% and 7.04%. Finally, based on f-measure, the developed scheme outperformed the existing schemes by 4.65% and 9.22%.

CHAPTER FIVE

SUMMARY, CONCLUSION, AND RECOMMENDATION

5.1 Summary

In this work, the development of an improved keyframe extraction scheme for video summarization based on histogram difference and k-means clustering has been presented. The developed scheme extracted a set of unique keyframes and eliminated duplicated ones, hence, improving the bandwidth utilization and storage capacity. Thus, redundant keyframes extracted due to the presence of visual editing effects such as gradual transitions, sudden illuminance, and camera movement are been reduced to the barest minimum.

5.2 Conclusion

The study presented the development of an improved keyframe extraction scheme for video summarization based on histogram difference and k-means clustering. The keyframe extraction scheme was developed by utilizing the histogram-based approach for shot boundary detection and the k-means clustering approach for the extraction of representative frames in the video shots. The algorithm was tested on four different videos (advert, movie clip, sport, and surveillance footage) downloaded from the popular video-sharing website YouTube. The performance of the developed scheme was compared with the existing scheme using the compression ratio, precision and extraction rates, f-measure. The developed scheme outperformed the existing schemes in terms of the compression ratio, precision and extraction rates, and f-measure.

5.3 Significant Contributions

The significant contributions of this study are as follows:

1. Development of an efficient k-means clustering-based approach of keyframe extraction for efficient video summarization.

2. The developed algorithm utilized a histogram-based technique to segment interrelated video frames into shot and k-means clustering to select a unique set of keyframes. Hence, improving the storage capacity of a device by 40%. Also, provides the user with a better experience in video browsing and retrieval.

3. The developed algorithm achieved an average of 9.22% improvement on f-measure when tested on the acquired videos. Also, it is more compressible when compared with the existing schemes.

5.4 Recommendations for Future Work

The following possible areas of further work are recommended for consideration for future research:

1. The study only considered 1080p HD videos downloaded from the Internet, local videos can be considered in future research.

2. The study can be further implemented in real-time applications.

REFERENCE

Abdulhussain, S. H., Ramli, A. R., Saripan, M. I., Mahmmod, B. M., Al-Haddad, S. A. R., & Jassim, W. (2018). Methods and challenges in shot boundary detection: a review. *Entropy. 20*(4), 214.

Coates, A., Ng, A. Y., & Lee, H. (2011). An analysis of single-layer networks in unsupervised feature learning. *International Conference on Artificial Intelligence and Statistics*. 215–223.

Adedokun, A. E., Abdulrazak, M. B., Momoh, M. O., Bello-Salau, H., & Sadiq, B. O. (2019). A Spatio-Temporal based Frame Indexing Algorithm for QoS Improvement in Live Low-Motion Video Streaming. *ATBU Journal of Science, Technology Education. 7*(3), 305-315.

Ali, I. H., & Al–Fatlawi, T. (2019). A Proposed Method for Key Frame Extraction. *International Journal of Engineering Technology*. *8*(15), 509-512.

Amiri, A., & Fathy, M. (2010). Video shot boundary detection using QR-decomposition and gaussian transition detection. *EURASIP Journal on Advances in Signal Processing. 2009*(1), 509438.

Asghar, M. N., Hussain, F., & Manton, R. (2014). Video indexing: a survey. *International Journal of Computer Information Technology*. *3*(01).

Asim, M., Almaadeed, N., Al-Máadeed, S., Bouridane, A., & Beghdadi, A. (2018). A key frame based video summarization using color features. *Paper presented at the 2018 Colour and Visual Computing Symposium (CVCS).*

Azeroual, A., Afdel, K., El Hajji, M., & Douzi, H. (2014). Video Shot Detection and Key Frame Extraction Using Faber Shauder DWT and SVD. *International Journal of Computer, Electrical, Automation, Control Information Engineering.*

Azhar, A. Z., Pramono, S., & Supriyanto, E. (2016). An Analysis of Quality of Service (QoS) In Live Video Streaming Using Evolved HSPA Network Media. *JAICT, 1*(1).

Benni, V., Dinesh, R., Punitha, P., & Rao, V. (2015). Keyframe extraction and shot boundary detection using eigen values. *International Journal of Information Electronics Engineering. 5*(1), 40.

Bhaumik, H., Bhattacharyya, S., Nath, M. D., & Chakraborty, S. (2016). Hybrid soft computing approaches to content based video retrieval: A brief review. *Applied Soft Computing. 46,* 1008-1029.

Bi, C., Yuan, Y., Zhang, J., Shi, Y., Xiang, Y., Wang, Y., & Zhang, R. (2018). Dynamic mode decomposition based video shot detection. *IEEE Access, 6*, 21397-21407.

Birinci, M., & Kiranyaz, S. (2014). A perceptual scheme for fully automatic video shot boundary detection. *signal processing: image communication. 29*(3), 410-423.

Cao, C., Chen, Z., Xie, G., & Lei, S. (2012). Key frame extraction based on frame blocks differential accumulation. *24th Chinese Control and Decision Conference,* 3621-3625. doi: 10.1109/CCDC.2012.6243092.

Cernekova, Z., Pitas, I., & Nikou, C. (2005). Information theory-based shot cut/fade detection and video summarization. *IEEE Transactions on circuits systems for video technology. 16*(1), 82-91.

Chen, Y., Deng, Y., Guo, Y., Wang, W., Zou, Y., & Wang, K. (2010). A temporal video segmentation and summary generation method based on shots' abrupt and gradual transition boundary detecting. *Paper presented at the 2010 Second International Conference on Communication Software and Networks*.

Choroś, K. (2011). Reduction of faulty detected shot cuts and cross dissolve effects in video segmentation process of different categories of digital videos. *International Transactions on computational collective intelligence V* (pp. 124-139): Springer.

Dailianas, A., Allen, R. B., & England, P. (1996). Comparison of automatic video segmentation algorithm*s. Paper presented at the Integration Issues in Large Commercial Media Delivery Systems.*

Del Fabro, M., & Böszörmenyi, L. (2013). State-of-the-art and future challenges in video scene detection: a survey. *Multimedia systems. 19*(5), 427-454.

Ejaz, N., Tariq, T. B., & Baik, S. W. (2012). Adaptive key frame extraction for video summarization using an aggregation mechanism. *Journal of Visual Communication Image Representation. 23*(7), 1031-1040.

Furini, M., Geraci, F., Montangero, M., & Pellegrini, M. (2010). STIMO: STIll and MOving video storyboard for the web scenario. *Multimedia Tools Applications. 46*(1), 47.

Gharbi, H., Bahroun, S., Massaoudi, M., & Zagrouba, E. (2017). Key frames extraction using graph modularity clustering for efficient video summarization. *Paper presented at the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* 1502-1506.

Gharbi, H., Bahroun, S., & Zagrouba, E. (2016). A Novel Key Frame Extraction Approach for Video Summarization. *Paper presented at the VISIGRAPP (3: VISAPP).*

Guo, Y., Xu, Q., Sun, S., Luo, X., & Sbert, M. (2016). Selecting video key frames based on relative entropy and the extreme studentized deviate test. *Entropy.*

Hu, J., Fang, L., Cao, Y., Zeng, H.-J., Li, H., Yang, Q., et al. (2008). Enhancing text clustering by leveraging Wikipedia semantics. In *Proceedings of the 31st Annual International ACM*

SIGIR Conference on Research and Development in Information Retrieval (pp. 179–186). New York, NY, USA: ACM.

Huang, C., & Wang, H. (2018). A Novel Key-frames Selection Framework for Comprehensive Video Summarization. *IEEE Transactions on Circuits and Systems for Video Technology*. doi:10.1109

Jadon, S., & Jasim, M. (2019). Video Summarization based on uniform sampling. *Paper presented at the International Conference on Image and Signal Processing*.

Jadhava, P., & Jadhav, D. (2015). Video summarization using higher order color moments. *Paper presented at the Proceedings of the International Conference on Advanced Computing Technologies and Applications (ICACTA).*

Janwe, N. J., & Bhoyar, K. K. (2013). Video shot boundary detection based on JND color histogram. *Paper presented at the 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013).*

Janwe, N. J., & Bhoyar, K. K. (2016). Video key-frame extraction using unsupervised clustering and mutual comparison. *International Journal of Image Processing. 10*(2), 73-84.

Jiang, X., Sun, T., Liu, J., Chao, J., & Zhang, W. (2013). An adaptive video shot segmentation scheme based on dual-detection model. *Neurocomputing. 116*, 102-111.

Kathiriya, Dhaval S. Pipalia, Gaurav B. Vasani, Alpesh J. Thesiya, & Varanva, D. J. (2013). X^2 (Chi-Square) Based Shot Boundary Detection and Key Frame Extraction for Video. *International Journal of Engineering and Science*, *2*(2), 17-21.

Kaur, P., & Kumar, R. (2018). Analysis of Video Summarization Techniques. *International Journal for Research in Applied Science & Engineering Technology (IJRASET), 6*(01).

Kavitha, J., & Rani, P. A. J. (2015). Static and multiresolution feature extraction for video summarization. *Procedia Computer Science 47*, 292-300.

Kawai, Y., Sumiyoshi, H., & Yagi, N. (2007). Shot Boundary Detection at TRECVID 2007. *Paper presented at the TRECVID.*

Küçüktunç, O., Güdükbay, U., & Ulusoy, Ö. (2010). Fuzzy color histogram-based video segmentation. *Computer Vision Image Understanding. 114*(1), 125-134.

Kumar, K., Shrimankar, D. D., & Singh, N. (2018). Eratosthenes sieve based key-frame extraction technique for event summarization in videos. *Multimedia Tools Applications. 77*(6), 7383-7404.

Lee, M. S., Yang, Y. M., & Lee, S. W. (2001). Automatic video parsing using shot boundary detection and camera operation analysis. *Pattern Recognition. 34*(3), 711-719.

Li, J., Yao, T., Ling, Q., & Mei, T. (2017). Detecting Shot Boundary with Sparse Coding for Video Summarization. *Neurocomputing*

Li, X., Zhao, B., & Lu, X. (2017). Key frame extraction in the summary space. *IEEE transactions on cybernetics, 48*(6), 1923-1934.

Ling, X., Yuanxin, O., Huan, L., & Zhang, X. (2008). A method for fast shot boundary detection based on SVM. *Paper presented at the 2008 Congress on Image and Signal Processing.*

Liu, C., Wang, D., Zhu, J., & Zhang, B. (2013). Learning a contextual multi-thread model for movie/tv scene segmentation. *IEEE transactions on multimedia. 15*(4), 884-897.

Liu, G., Wen, X., Zheng, W., & He, P. (2009). Shot boundary detection and keyframe extraction based on scale invariant feature transform. *Paper presented at the 2009 Eighth IEEE/ACIS International Conference on Computer and Information Science.*

Lu, Z.-M., & Shi, Y. (2013). Fast video shot boundary detection based on SVD and pattern matching. *IEEE Transactions on Image processing. 22*(12), 5136-5145.

Lv, C., & Huang, Y. (2018). Effective Keyframe Extraction from Personal Video by Using Nearest Neighbor Clustering. *Paper presented at the 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI).*

Mithlesh, C. S., & Shukla, D. (2016). A Case Study of Key frame Extraction Techniques. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 5*(3), 1292-1298. doi:10.15662/IJAREEIE.2016.0503016

Murugan, A. S., Devi, K. S., Sivaranjani, A., & Srinivasan, P. (2018). A study on various methods used for video summarization and moving object detection for video surveillance applications. *Multimedia Tools Applications.*

Olaniyi, S. B. (2014). Development of a Matlab Guided Based Interactive Platform for Edge Detection in Noisy Coloured Images. *International Journal of Engineering Research & Technology (IJERT), 3*(11), 66-69.

Paul, A., Milan, K., Kavitha, J., Rani, J., & Arockia, P. (2018). Key-Frame Extraction Techniques: A Review. *Recent Patents on Computer Science. 11*(1), 3-16.

Priya, G. L., & Domnic, S. (2014). Shot boundary-based keyframe extraction for video summarisation. *International Journal of Computational Intelligence Studies. 3*(2-3), 157-175.

Raikwar, S. C., Bhatnagar, C., & Jalal A. S. (2014). A frame work for key-frame extraction from surveillance Video. *Paper presented at the 2014 fifth IEEE International Conference on Computer and Communication Technology (ICCCT).* 297-300. doi:10.1109/ICCCT.2014.7001508.

Rashmi, B., & Nagendraswamy, H. (2016). Shot-based keyframe extraction using bitwise-XOR dissimilarity approach. *Paper presented at the International Conference on Recent Trends in Image Processing and Pattern Recognition.*

Ren, L., Qu, Z., Niu, W., Niu, C., & Cao,Y. (2010). Key frame extraction based on information entropy and edge matching rate. *Paper Presented at the 2010 Second IEEE International Conference on Future Computer and Communication (ICFCC).*

Rodriguez, J. M. D., Yao, P., & Wan, W. (2018). Selection of Key Frames Through the Analysis and Calculation of the Absolute Difference of Histograms. *Paper presented at the 2018 International Conference on Audio, Language and Image Processing (ICALIP).*

Salehin, M. M., & Paul, M. (2017). A novel framework for video summarization based on smooth pursuit information from eye tracker data. *Paper presented at the 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW).*

Santini, S. (2007). Who needs video summarization anyway? *Paper presented at the International Conference on Semantic Computing (ICSC 2007).*

Satpute, A. M., & Khandarkar, K. R. (2017). Video Summarization by Removing Duplicate Frames from Surveillance Video Using Keyframe Extraction. *International Journal of Innovative Research in Computer and Communication Engineering*, 8501-8509. doi:10.15680/IJIRCCE.2017. 0504265

Sheena, C. V., & Narayanan, N. (2015). Key-frame extraction by analysis of histograms of video frames using statistical methods. *Procedia Computer Science. 70*, 36-40.

Srinivas, M., Pai, M. M., & Pai, R. M. (2016). An improved algorithm for video summarization–a rank based approach. *Procedia Computer Science*

Sujatha, C., & Mudenagudi, U. (2011). A study on keyframe extraction methods for video summary. *Paper presented at the 2011 International Conference on Computational Intelligence and Communication Networks.*

Tapu, R., & Zaharia, T. (2011). Video segmentation and structuring for indexing applications. *International Journal of Multimedia Data Engineering Management. 2*(4), 38-58.

Tirupathamma, S. M. (2017). Key frame based video summarization using frame difference. *International Journal of Innovative Computer Science & Engineering, 4*(03), 160-165.

Tong, W., Song, L., Yang, X., Qu, H., & Xie, R. (2015). CNN-based shot boundary detection and video annotation. *Paper presented at the 2015 IEEE international symposium on broadband multimedia systems and broadcasting.*

Wang, D., Liu, J., Sun, J., Liu, W., & Li, Y. (2012). A novel key-frame extraction method for semi-automatic 2D-to-3D video conversion. *Paper presented at the IEEE international Symposium on Broadband Multimedia Systems and Broadcasting.*

Wu, J., Zhong, S.-h., Jiang, J., & Yang, Y. (2017). A novel clustering method for static video summarization. *Multimedia Tools Applications, 76*(7), 9625-9641.

Wu, K. (2011). Simple Implementations of Video Segmentation, Key Frame Extraction and Browsing.

Yuan, J., Wang, H., Xiao, L., Zheng, W., Li, J., & Lin, F. (2007). A formal study of shot boundary detection. *IEEE transactions on circuits systems for video technology. 17*(2), 168-186.

Yuan, J., Wang, W., Yang, W., & Zhang, M. (2014). Keyframe extraction using AdaBoost. *Paper presented at the Proceedings 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC).*

Zedan, I. A., Elsayed, K. M., & Emary, E. (2018). News Videos Segmentation Using Dominant Colors Representation. In *Advances in Soft Computing and Machine Learning in Image Processing* (pp. 89-109): Springer.

Zhang, Q., Yu, S.-P., Zhou, D. S., & Wei, X. P. (2013). An efficient method of key-frame extraction based on a cluster algorithm. *Journal of human kinetics. 39*(1), 5-14.

APPENDIX A

MATLAB CODE FOR SHOT BOUNDARY DETECTION

```
clear all;
clc;
tic;
MOV = VideoReader('advert.avi');
NumberOfFrame = MOV.NumberOfFrames;
data = zeros(fix(NumberOfFrame/w),4);
DataDi = zeros(fix(NumberOfFrame/w),(w+1));
shotset = [0];
%frame diff
for k=1:fix(NumberOfFrame/w)
    for i=1:w
        Frame_1 = read(MOV,(k-1)*w+i);
        Frame_2 = read(MOV,(k-1)*w+i+1);
        D(i) = FrameDiff(Frame_1,Frame_2);
        DataDi(k,1) = k;
        DataDi(k,i+1) = D(i);
    end
%threshold
    [threshold_h,threshold_l]=ThresholdCounting(w,D);
    data(k,1) = k;
    data(k,2) = threshold_l;
    data(k,3) = threshold_h;
  h=stem(D);
   saveas(h,strcat('C:/Documents/MATLAB/KeyFrameExtraction/shot',num2str(k),'.bmp'),'bmp');
    GradationMark = 0;
    tolerance = 0;
  if(threshold_h<1)
        FrameId = find(D==max(D))+w*(k-1);
        shotset = union(shotset,[FrameId]);
        data(k,4) = FrameId;
        clear FrameId;
    else
 for j=1:w
            FrameId = (k-1)*w+j;
            if(D(j)>threshold_l)
                if(GradationMark==0)
                    GradationMark = 1;
                end
            end
        end
    end
```

APPENDIX B

MATLAB CODE FOR THE DEVELOPED SCHEME

```
NumberOfShot = length(shotset);
%segmenting the video frames into shots
for k=1:(NumberOfShot-1)
    F = read(MOV,shotset(k)+1);
    NumberOfCluster = 1;
    Cluster(1).center = FeatureOfImg(F);
    Cluster(1).number = 1;
    Cluster(1).img = cell(2,1);     %
    Cluster(1).img{1,1} = F;
    Cluster(1).img{2,1} = FeatureOfImg(F);
for i=(shotset(k)+1):shotset(k+1)     %
    img = read(MOV,i);
    imgfeature = FeatureOfImg(img);
    for j=1:NumberOfCluster
        Dis(j) = norm(imgfeature-Cluster(j).center);   %
    end
        IdOfNearestCluster = find(Dis==min(Dis));
Cluster(IdOfNearestCluster).center =
Cluster(IdOfNearestCluster).center*Cluster(IdOfNearestCluster).number/(Cluster(IdOfNearestCl
uster).number+1)+imgfeature/(Cluster(IdOfNearestCluster).number+1);
            Cluster(IdOfNearestCluster).number = Cluster(IdOfNearestCluster).number+1;
            Cluster(IdOfNearestCluster).img{1,Cluster(IdOfNearestCluster).number} = img;
            Cluster(IdOfNearestCluster).img{2,Cluster(IdOfNearestCluster).number} = imgfeature;
    end
end
clear Dis;
% key frame extraction
for i=1:NumberOfCluster
    if(Cluster(i).number>(shotset(k+1)-shotset(k))/NumberOfCluster)
        for j=1:Cluster(i).number
            D(j) = norm(Cluster(i).img{2,j}-Cluster(i).center);
        end
        IdOfKeyFame = find(D==min(D));
        Keyframe = Cluster(i).img{1,IdOfKeyFame};
            imwrite(Keyframe,strcat('C:/Users/BSMUHD/Documents/MATLAB/KeyFrame
project/keyframe1/shot',num2str(k),'cluster',num2str(i),'.jpg'),'jpg');
    else
    end
    clear D;
end
```