Keyframe-Based Saliency Detection for Human Action Recognition Using Deep Learning

By

Diriba Aso Chewaka

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University

October 2020

Adama, Ethiopia

Keyframe-Based Saliency Detection for Human Action Recognition Using Deep Learning

By

Diriba Aso Chewaka

Advisor: Worku Jifara (Ph.D.)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Office of Graduate Studies

Adama Science and Technology University

October 2020

Adama, Ethiopia

DECLARATION

I intensely state that this thesis is my unique work and it is not has been submitted on the other degree or diploma of other universities. Similarly, it had been not presented as a partial degree requirement for a degree in other universities. All the sources and materials used are acknowledged.

Name: Diriba Aso

Signature: _____

This MSc. thesis has been submitted for examination with my approval as a thesis advisor.

Name: Worku Jifara (Ph.D.)

Signature: _____

Date of submission: _____

APPROVAL PAGE

We, the undersigned, members of the board of examiners of the final open defense by "*Diriba Aso Chewaka*" have read and evaluated his thesis entitled "*Keyframe-Based Saliency Detection for Human Action Recognition Using Deep Learning*", examined the candidate. This is therefore to certify that the thesis has been accepted in partial fullfilment of the requirement of degree of Masters of Science in Computer Science and Engineering.

_____	_____	_____
Student Name	*Signature*	*Date*
_____	_____	_____
Advisor	*Signature*	*Date*
_____	_____	_____
Chairperson	*Signature*	*Date*
_____	_____	_____
External Examiner	*Signature*	*Date*
_____	_____	_____
Internal Examiner	*Signature*	*Date*
_____	_____	_____
Department Head	*Signature*	*Date*
_____	_____	_____
School Dean	*Signature*	*Date*
_____	_____	_____
Post Graduate Dean	*Signature*	*Date*

ACKNOWLEDGMENT

Whatever my life journey, first and foremost, I want to thank my God for all that he has done and planned for me including this level of study as one of my pathways.

Next, I strongly desire to express my deep gratitude to Worku Jifara (Ph.D.) for his advising, kindly approach and support to complete this thesis and Yun Koo Chung (Ph.D.), who plays a major role in my SIG by facilitating accessible resources as well as sharing knowledge, and Mesfin Abebe (Ph.D.).

My special thanks were for all of my SIG members and Mr. Anteneh Tilahe.

Finally, I want to say God bless you my family as well as Kebede Shiferaw, the base initiator for my education journey.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

AR	Action Recognition
ASTU	Adama Science and Technology University
AUC	Area Under Curve
AVI	Audio Video Interleaved
BOW	Bag of Words
CNN	Convolutional Neural Network
CONV-LSTM	Convolutional Long Short-Term Memory
ConvNet	Convolutional Network
CPU	Central Processing Unit
DL	Deep Learning
DLHAR	Deep Learning-Based Human Action Recognition
HAR	Human Action Recognition
GLOH	Gradient Location Oriented Histogram
GRNN	Gated Recurrent Neural Network
GRU	Gated Recurrent Unite
GPU	Graphical Processing Unit
HD	High Definition
HOOF	Histogram of Oriented Flow
HOG	Histogram of Gradient
IDT	Improved Dense Trajectory
KF	Kalman Filter
KFBSD	Key Frame-Based Saliency Detection
KFS	Key Frame Selection
KFSSD	Key Frame Selection Saliency Detection
KFSD DLBHAR	Key Frame Saliency Detection for Deep Learning-Based Human Action Recognition
LBP	Local Binary Point
MHI	Motion History Image
OMCNN	Object Motion Convolutional Neural Network
OpenCV	Open Source Computer Vision
PAGNet	Pyramidal Guided Attention Network

ReLU	Rectified Linear Unit
RCB	Region Contrast Boundary
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristics
RPCA	Robust Principal Component Analysis
RQ	Research Question
SD	Saliency Detection
SIFT	Scale Invariant Feature Transform
SIG	Special Interest Group
SDN	Saliency Detection Network
SS-CONVLSTM	Saliency Structured Convolutional Long Short-Term Memory
SHI	Static History Image
SMHI	Structural Motion History Image
SVM	Support Vector Machine
UCF	University of Central Florida
UML	Unified Modeling Language
VGG	Visual Geometry Group
VS	Video Saliency

ABSTRACT

Human action recognition is an active research area in the computer vision and it is widely applied to video processing, surveillance, home automation, healthcare, human behavior analysis, etc. Problems such as the inclusiveness of irrelevant features and redundant video frames affects the performance of action recognition. To overcome such challenges, traditional methods for recognizing human actions where the main focus was on body motions were studied. With the recent advances in deep learning, techniques that integrate spatial and temporal features with reduced dimension have been developed with marked success over the traditional methods. However, solely extracting the informative frames for representing action, and including only relevant information for fast processing remains a challenge. Therefore, in this study, keyframe-based saliency detection using deep learning was developed to overcome the stated problems. This study is aimed at developing keyframe-based saliency detection to recognize human actions using deep learning. Keyframe selection using color difference of consecutive frames compared to local maximum, saliency detection using pyramidal attention guided network, and VGG-16+Bi-directional LSTM for action recognition were used. UCF101 dataset of 20 classes was used for training and validating the network using 70%, 30% train-test split approach. By applying the developed KFSD HAR model, the overall training accuracy of 86% and 61sec time *was achieved. Generally, the developed KFSD HAR outperforms existing VGG-Bi-LSTM, VGG-LSTM and CNN models having training accuracy (4.023%, 53.2%, 83%) and time(329 sec, 175 sec., 66 sec.) respectively due to potential keyframes selected from video and detected using saliency method for action recognition.*

Keywords: *Human Action Recognition, Keyframe Selection, Saliency Detection, Video Analysis, KFSD, Keyframe, Saliency*

CHAPTER ONE

1. INTRODUCTION

1.1 Background of the Study

Video analysis is the subject area of computer vision that involves automatic interpretation of digital videos by using computer algorithms to gauge concepts and result from shown content [1]. Additionally, it is the procedure of using any motion recording, sequences of images performing some action to understand actionable information. According to [2], in recent years, the development of digital videos increased and tends to provide informative content. This resulted from its wide applicability in different areas such as distance learning, tourism, bus station, advertisement, airport, sports center, supermarket, health center, traffic station, etc. Such applications include video content retrieval and search, detection of an anomaly, autonomous driving, object tracking, and human action recognition [3].

From the video analysis applications, human action recognition, however, was studied in the late 19th century [3]. At that time, it has been conducted to serve human life as arbitrary and application-oriented wearable sensors. Continuously, in the late 20th century, it was used for assisting human beings to adopt healthy lifestyles like brushing teeth, taking baths, and related activities [3]. Today, it has been applicable for most real-world situations and areas like surveillance, security, human-computer interaction, video annotation, content retrieval, robotics, ambient assisted living, police investigation, entertainment, healthcare system, fraud detection, human behavior analysis, mass population statistics as well as testing [4].

Moreover, for the future and even a little bit today, it has wide applicability to accomplish recognition tasks independently of the device and environment [4]. To facilitate these tasks, applications of technologies get major considerations. Some of these applications include Artificial intelligence, Computer Vision, Smart Security, Cloud Source, Advanced Machine Learning, etc. Today, the power of Computer Vision, as well as Robotics, spreads incrementally in many areas of studies including satellite imagery, biometric recognition, security surveillance, activity recognition, behavioral analysis, medical diagnosis, and surgery [5].

Computer vision plays a great role in activity recognition especially human action recognition [6]. However, most algorithms and studies were related to motion analysis which

is resulted from body structures. This body information is selected by wearable sensor devices. Current studies were more focused on effective algorithms to understand the accurate recognition of humans in the field of Advanced computer vision, Machine Learning, and Deep Learning. Even if these algorithms were targeted to achieve recognized actions of humans, their level of ability to recognize is not the same. Beyond this, the power of video analyses by current methods especially deep learning relies on efficient resources such as High Processing Computers, very large data, High-speed processors such as GPU, and smart integrated sensors. Accessing all these for a single application at one time was not always promising. Additionally, the problem of computational time and accuracy was another issue to be considered.

However, many researchers have been studied human action recognitions such as vision-based human action recognition in surveillance videos using motion projection profile which used Motion Projection Profile for feature selection and Gaussian Mixture Model for classifier [7], Automated daily human activity recognition for video surveillance using the neural network which used Digital Image Processing for feature selection and Multi-Layer Forward Perceptron as classifier [8], deep learning approach for human activity recognition in a smart home [9] which used CNN for feature selection and SoftMax as a classifier to overcome some of the stated problems.

Many studies have been conducted using traditional methods such as Human action recognition using Discrete Fourier Transform (DFT) [19], Human Action Recognition Using Improved Salient Dense Trajectories (IDT) [21], A Comprehensive Survey of Vision-Based Human Action Recognition Methods [22], A survey on still image-based human action recognition [24], etc. concerning the aforementioned tasks. Additionally, by deep learning technique like as Recognition of actions in sequences of video based on Bi-Directional LSTM which is Deep with features of Convolutional Neural Network [18], a hybrid model of deep learning for human action recognition which is a new method [32], Detection of action by using Neural Networks which is Deep: problems as well as solutions[33], Sequential Deep Learning for Human Action Recognition in Human Behavior Understanding [34], there has been much research conducted and under investigation. Compared to the traditional methods for recognizing human action, the deep learning methods were far better realistic for the achieved result.

However, most of the deep learning-based approach has been using background subtraction in their starting steps of the recognition process. But the background subtraction may not produce the expected results since it needs accurate object detection and the best classification model for classifying pixels as background and foreground [13].

Additionally, the feature selection problem may exist that could have resulted from inappropriate data augmentation (pre-processing step). This means, most of the human action recognition using deep learning-based has been using real data with their scenes in their begging of action recognition steps and background subtraction. This was sometimes challenging if the scene contains multi-object per data, a fast object moving, camera motion, background and foreground color or object similarity and background clutter. Moreover, the deep learning approaches that used RGB data with depth information may require much inclusive information that can cause more hard-working for feature selection and processing.

Additionally, in video sequences, not all stacked frames were needed to decide the action. This means, some frames may be relevant to represent the action but, others were very related together and are almost the same which may cause frame redundancy and a lot of computational time.

To overcome such challenges, some works have been conducted such as Temporal Segmentation Network**,** Keyframe selection from frame sequences [11]. For the problems foreground-background separation in a complex scene, spatial-temporal salience detection[12], Motion dominated Fast Object Segmentation [13], Appearance dominated robust principal component analysis [14], fusing disparate object signature for salient object detection in the video [15]**,** video background selection in the complex environment [16] were studied.

In this study, the task of recognition of human action was conducted based on the approach of selecting the relevant feature of the keyframes in a sequence to decide the action to be processed. The method includes (1) only selected frames, i.e. keyframes per video action clips were selected for starting the recognition process to choose frames those have potential information for representing the action, (2) the selected frames were detected using methods of saliency detection to include only relevant object information per scene and (3) only selected keyframe which was detected based on their salient information were used for action recognition step using VGG-16 + bi-directional LSTM.

Generally, the study was focused on key frame-based saliency detection for recognition of human action and examining as well as evaluating its performance.

1.2 Motivation of the Study

Video analysis was one of the current hot issues mostly in computer vision as well as robotics subjects. This is due to its wide applicability in different areas such as ambient human assistive living, surveillance, object tracking, video content retrieval, and human action recognition for solving challenges related to the need of performing video-based tasks such as forensics or investigation, ease of treatment in health care services, supporting disable and special support need peoples.

As one application of video analysis, human action recognition focused on understanding the action of the person from the sequence of motion analyzed from the video. For performing the recognition tasks, researchers have been studying computer vision and deep learning methods besides the machine learning approaches. Compared to computer vision and other machine learning approaches for human action recognition, deep learning outperforms the task much better. This can be shown from multi-feature selection and learning as well as a better result at the end of the process. However, some of the current human action recognition based on deep learning-based has been using real data with information such as object and its background, additional or irrelevant information per scene as well as interference. This requires long processing time and storage. Additionally, recognition results may be inaccurate since there may be irrelevant information selected and learned in the process. If such challenges were solved, human action recognition can be accurate and fast process.

Therefore, in this study key-frame based saliency detection for human action recognition was developed to overcome most existing problems of deep learning-based human action recognitions such as redundant input data, i.e. video frames per action, irrelevant or additional information to the data, i.e. background interference and clutter.

The method, keyframe based saliency detection for human action recognition was preferred due to (1) only selected frames, i.e. keyframes per video action clips were selected for starting the recognition process to choose keyframes those have potential information for representing the action, (2) the selected keyframes were detected using saliency detection method to include only relevant object’s information per scene and, (3) only selected keyframes which are detected based on the salient information was used for recognition step

using VGG-16 + bi-directional LSTM, that is a type recurrent neural network for learning sequential information in two directions (one from past and one from the present state of the action). By doing so, the process of human action recognition was fast and requires less effort.

1.3 Statement of the Problems

Even if human action recognition gets much attention in video analysis applications, there is a huge gap between attained goals and expected needs. For instance, fast processing, better accuracy, the use of real data having a heterogeneous background, camera motion, occlusion, and clutter or interference compared to get a better result.

Although most of such challenges were solved in deep learning, still it needs properly handled data. The problem of using only selecting potential frames per action and including only relevant information for the recognition process still needs further study.

Moreover, the major problems identified in this proposed work were the inclusiveness of irrelevant features which requires computational time, dense sampling every frame which may cause large computation cost and frame redundancy, sparse sampling which may cause loss of information especially in fast motion, action recognition challenge due to some similarity between background and objects, drastic illumination change, video scene understanding with a complex background, object detection and feature extraction.

In this study, the problem of fame redundancy in video and including relevant feature were addressed by selecing potential frames from video and detecting the salient informations of the selected keyframes.

The result of selected keyframes and detected salient information of keyframes have rich information to represent the action in less frame and relevant feature. This is due to the keyframe selection and saliency detection focuses on selecting only frames which have potential information to represent the other and only important regions to represent the actions were detected using saliency detection.

Detecting only salient region enhances the extraction of essential features from frames and discarding the irrelevant informations such as drastic illumination change, background clutter, and background complexity per scene.

Generally, in this study the problem of having redundant or duplicated frame while recognizing an action and complexity of background, clutter and camera motion were solved

using keyframe selection and saliency detection method.This method enhances the fast processing for action recognition and accuracy.

1.4 Research Questions

In this study, the potential challenges of existing human action recognition techniques or methods were studied. Even if there were different questions regarding the human action recognition method today, two major questions which are stated as RQ1, RQ2 and RQ3 were identified. These questions are:

RQ1: What is the current challenges of sequence-based human action recognition?

RQ2: How to overcome video frame redundancy and irrelevant feature inclusiveness problems from video sequences?

RQ3: What is the effect of applying key frame-based saliency detection on human action recognition?

By understanding the major problems in existing human action recognition [56], these three questions were selected as the core issue and studied in this work.

1.5 Objectives of the Study

1.5.1 General Objective

The general objective of this study is to develop keyframe-based saliency detection for human action recognition using deep learning.

1.5.2 Specific Objectives

To achieve the general objective, some specific objectives were identified. These includes:

- To understand sequence-based human action recognition challenges in detail by reviewing journals, magazines, workshops and tutorials.
- To develop an algorithm for selecting of keyframe based on the RGB difference of consecutive frames, computing local maximum and comparing.
- To construct a saliency-based action detection model using a pyramidal guided attention network.
- To apply keyframe-based saliency detection for including action's relevant information.
- To implement the developed keyframe-based saliency detection for human action recognition using VGG-16 + Bi-directional LSTM.

- To evaluate the performance of the developed keyframe-based saliency detection for human action recognition model using accuracy for training, validation, and prediction , loss for training and validation.

1.6 Scope and Limitation of the Study

1.6.1 Scope of the Study

Since human action recognition is a wide area of study and there are many methods applied, it is not possible to generalize and cover all concerning tasks in all dynamic environment. Based on this concept, the main focus of this work is to develop keyframe based saliency detection and apply on existing human action recognition using VGG-16 + Bi-Directional LSTM, then finally comparing selected existing deep learning-based recognition of human action with newly developed KFSD HAR.

Additionally, measuring the effect of applying keyframe based saliency detection on existing human action recognition using VGG-16 + bi-directional LSTM is considered. To do this task, keyframe selection and saliency detection algorithms were developed and implemented. For the developed keyframe selection, the main difference from existing algorithm is that (1) reading multiple input data files and produce the out in which the the existing reads only one data at a tme.(2) The output achieved were as expected compared to existing keyframe selection which deviates from the correct assumptions. For instance a video of 6 sec. stored at 25 frame/sec would have 25*6 = 150 frame sequence. From the sequences, the existing keyframe selection uses only two keyframes and discards 148 frames. But in the proposed, 100 keyframes were selected. (3) the mechanism of sequencing selected keyframes. Since the action recognition network(VGG-16+Bi-LSTM) works based on sequences of frames, this stage is very crucial in which existing algorithm doesnot focus.

In the developed saliency detection, the mechanisms of input and output, the improvement of saliency results and conversion of the result to expected input for action recognition were the major contributions. Therefore, the scope of this study includes:

- Human action recognition for selected five types of actions (human-human, human-object, body motion only, sports and playing musical instruments).
- Selecting keyframes from human action video using a keyframe selection algorithm.
- Saliency detection for selected keyframes to include relevant features.
- Applying developed keyframe based saliency detection on human action recognition using VGG-16 + Bi-directional LSTM.

- ❖ Evaluating the performance of the developed model using accuracy and loss for both training and validation.
- ❖ Comparing the developed model result with existing human action recognition models such as CNN, VGG-LSTM, and VGG-Bi-LSTM using accuracy, loss and time.

Activities such as studying nature, the behavior of the deep learning system, evaluation criteria beyond accuracy and loss were not included in this work. In this stsudy, recognition of human action beyond the selected human-human interactions, human-object interactions, body motion only, sports, and playing musical instrument actions was not included.

Other information of action in video sequences such as where the object is found, person identification, prediction of environment condition and actions as well as methods beyond the proposed keyframe-based saliency detection for human action recognition using VGG-16 + bi-directional LSTM for implementing the system were beyond the scope. Since the nature of the proposed work is based on frame selection, saliency detection, and rearranging the result of keyframe based saliency detection, offline implementation is provided. So, real-time or online processing is not included.

1.6.2 Limitations of the Study

The limitations proposed keyframe-based saliency detection for human action recognition includes: the timely accessibility of all resources required for the proposed system and lack of locally collected data except the online repository dataset used. Additionally, since the developed model requires a huge amount of all the data, processing all the data with available time and resources was inefficient. Lack of video level saliency detection and real-time recognition are also limitations of this work.

Since the focus is on recognizing what type of action a person shows in the sequence, there are no other required recognition results such as gender, age, number, appearance, dressing styles, etc. Additionally, for timely access to limited data for the system in a selected environment with device capability, the implemented system does not encompass all actions of the human beings over the world with large collected data from real-world problems. To take advantage of the data-hungry of the deep learning problem, 3D data was not used.

Moreover, since the available data is very challenging, and most of the actions were irregularly patterned, multi- visual similarity, the prediction of action result obtained by

implementing the proposed keyframe based saliency detection is less than the original VGG-16 + Bi-directional LSTM result. This shows saliency the implemented detection was not enough for such challenging data for better results.

Generally, recognition of actions beyond selected five types stated in the scope section, online action recognition, video level saliency detection, people detection and identification, predicting the future actions are the limitations of this study.

1.7 Beneficiary of the Study

This research has importance both in academics as well as solving human problems related and studied by using technological advancements. Even if the system is computerized, including stakeholder is useful for processing tasks which require a manual system. The beneficiaries of the study were categorized in different applicability includes:

1.7.1 Ambient Assistive Living

In this type of application of HAR, home services are given for special support holders which include: disable person., children, elders, etc.

In-home applications such users were the top beneficiaries of HAR because, by using a developed system, it is easy to identify and treat or support using what type of action or condition users were in. For instance, if an old man wants to drink water but can't due to some weakness or other difficulty, others can understand and provide the service. This is similar for disabled people and children.

It is known that babies are taking different things to their mouth and may catch danger such as fire, electricity, poisonous things, crewel to restricted places such as a hole. So that it can be easy to treat such support need users with the developed system.

1.7.2 Sport and Recreational Activities

In this type of application of HAR, the beneficiaries were sports manager or coach, Sports members. By using a developed system, it possible to manage sports activities that the members are performing without direct contact with the manager or coach.

1.7.3 Health

For such type of application of HAR, doctors or nurses can easily treat their patients by looking at the action of a person. For instance, if e-health is provided for patients at his/her home by doctors or nurses, only action information of the patient is delivered to the stated

doctor or nurse. So as required, doctors or nurses can provide health services for patients or attend and guide efficiently.

1.7.4 Video Processing

Video specialists working on video processing activities can benefit from such an application. Compression, summarization, video action detection, video content retrieval, etc. are made to be simple.

1.7.5 Security

Police or responsible persons for protecting criminal activities can be benefited from the system. This includes attending and protecting criminal acts that can manage or control wrong activities permed or performing by individuals. For instance, in-home, if someone eats poisoned food or drink, it is possible to investigate who did what.

Additionally, forensics, fraud detection, and plausible judgment were performed by the help of such security-based application of human action recognition. But for this implementation, if it is online-based, real-time protection can be performed which is immediate action and decision taken for respecting the rule of law.

1.8 Major Contrubutions and Achieved Outcomes

In this work, the idea of using only potential frames and including only relevant features for developing human action recognition through deep learning is performed. Additionally, for the developed model, contribution was made to each algorithms. For instance, nature of input data, i.e Channel information, type, number of data to be processed at a time, method of reading the input data to be processed, changing default values of parameters used and number of output after process were studied and applied.

Moreover, data processing and procedures was applied before they were directly used for implementation. By applying these contributions, the final outcomes includes: potential and representative frames were selected, relevant feature of selected frame was used which enhances feature extraction and processing time, improved accuracy and reduced time in training and validation were achieved.

Generally, in this work there is an emproved processing time, accuracy due to the developed keyframe-based saliency detection for human action recognition which uses action videos constructed from keyframes and detected saliency instead of using all video frames with

multi-informations such as background clutter, illumination changes and multi-object per scene.

1.9 Application of the Result

It is clear that HAR is almost generic and widely applied in the areas like automatic surveillance, security, video annotation and retrieval, robotics, police investigation, ambient assisted living, entertainment, fraud detection, etc. In the same manner, the proposed system successfully plays a great role in the stated applications, however, its main focus in this proposed work is for human action recognition from video.

Furthermore, the proposed system was also applicable for the detection of a salient object in the video, video keyframe selection, human activity recognition, ambient assistive living recognition, video object detection, video content retrieval, visual surveillance, smart rehabilitation, video frame segmentation, etc.

1.10 Organization of the Thesis

Even if it is not fully standardized format to the universal world, this study follows formats and standards drafted by ASTU Guideline for Postgraduate Studies.

Arrangement of this thesis is based on categorizing parts into chapters titles, subtitles, and sub-sub titles. Based on these concepts, the arrangement was as follows as acknowledgment, abstract, acronyms were given in preliminary parts.

Next, the main body parts include introduction written in chapter one, Literature Review and Related work in chapter two, research methodology in chapter three, the proposed keyframe-based saliency detection for human action recognition approach in chapter four, implementation of the proposed approach in chapter five, result and discussion in chapter six and conclusion and future work in chapter seven. Finally, references and appendix and sample of code were written at the last section.

1.11 Summary

In this chapter, the brief explanation of human action recognition by using different methods, the background, strengthens and weakness of each method, problems, objectives and methodology, limitations, beneficiaries, and major application areas were discussed. The study explains that the application of developed methodology can be evaluated whether or not it made a change to existing human action recognition using deep learning. Moreover, challenges and solutions to be undertaken to overcome were described.

CHAPTER TWO

2. LITERATURE REVIEW AND RELATED WORKS

2.1 Introduction

This chapter explains deep understating of existing human action recognition, challenges, methods, results, limitations, identified gaps to be fulfilled, major application areas, and some related works. The overall aim of this chapter is to establish a deep understanding of the area of study and identify major strengths and weaknesses to fill the gap existing system by applying the proposed method. Although the review may cover a variety of theories and concepts, its focus was the study of existing systems, problems or challenges and gaps to be fulfilled.

2.2 Human Action Recognition Overview

Recognizing the action of humans is the process of locating and understating of what someone is doing and recognition of action is the task of inferring actions of humans (presenting states) based on the completed execution of the action [17]. While human activity is a movement or motion of body parts that may or may not include interaction with objects in an environment. In the case of videos, action can be represented with frame sequences so that someone can simply understand by analyzing multi-frame sequences [18].

Action is every meaningful movement of a human being which convey important information and interact with the natural environment without every mechanical device [19]. It is known that every human action, whatever it is important is done for some roles. For instance, patient exercise using body parts like hand, leg, torso, etc. by interacting with the environments. Such actions can be observed by eyes or evaluated by visual sensors [17].

2.3 Human Action Recognition in Different Studies

2.3.1 Computer Vision

Computer vision is a field of study (interdisciplinary) or area which addresses how information which enriched is provided to achieve understanding videos or digital images at a high level [20]. The term human action studied in computer vision compasses from limp movement to body parts. Human action recognition gets attention as a hot issue in computer vision because of its applications in many real-world situations like human pose estimation, human detection, tracking, video content retrieval, surveillance, human-computer

interactions, and ambient assistive living [21]. Figure 2.1 shows the simple workflow of human action recognition using computer vision.

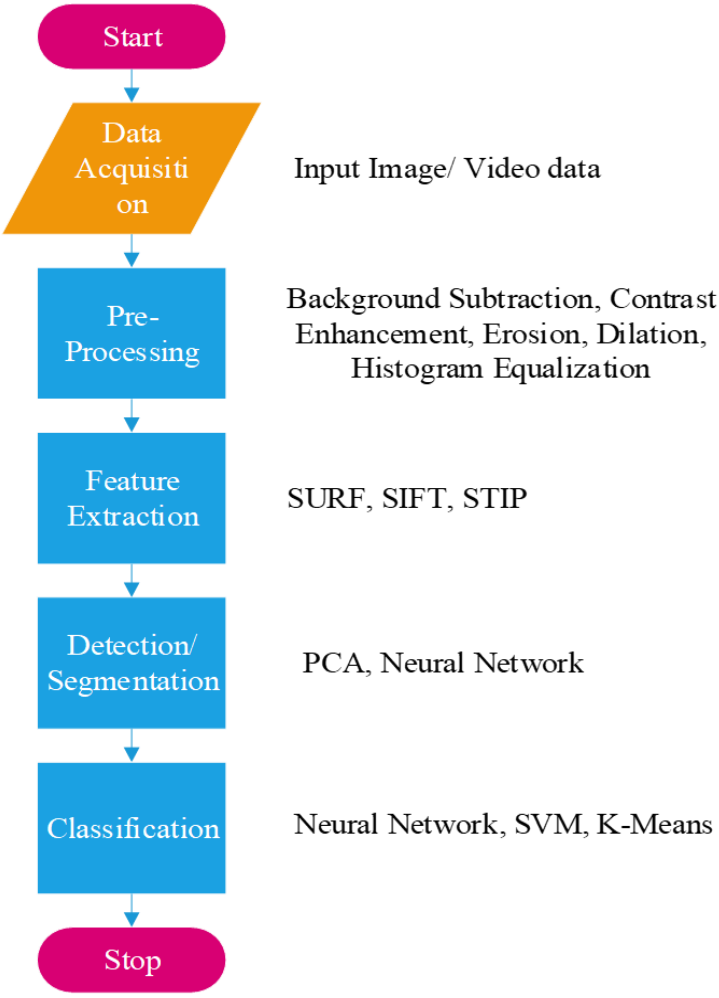


Figure 2.1 : Simple Workflow in Vision-based HAR

Source: adapted from [22]

From, the series of common workflows followed for recognizing human action using computer vision was described. First, input data which may be video or image based on the nature of the developed network is used. Next, preprocessing the input data is performed. Then extracting feature features from preprocessed data is conducted. Finally, for better to classify the extracted features, detection, or segmentation was applied and classify the action using preferred classifiers.

A. Recognizing Human Action in Computer Vision and its Problem

Even if a large number of algorithms, methods, and models are developed, human action recognition is still not much succeeded in the case of computer vision. This can be shown in

a case where some features used for recognitions are mostly defective and redundant due to variability, background interference, high complexity [21]. Additionally, in the case of video sequences, it is challenging for the reason that the visual content similarities, actions viewpoint change, the motion of the camera with action performer, actor's pose and scale, different illumination conditions [18].

Moreover, it is known that human action recognition in its good manner is that modeling the robust action of humans and representing the features. Different from representing features in image space, human action in a video is not only describe appearance but also appearance change selection which means spatial and temporal representation [22].

However, most recent state-of-the-art approaches of human action recognition use handcrafted features which in some cases use texture features and motions calculated based on STIP which is manually built. Even if such algorithms automatically extract features and achieves high performance, mostly they are problem-dependent. This means that most approaches of human action recognitions were used for the specific problem which shows some drawbacks.

Generally, clear problems represented in computer vision but not limed to these were: redundancy of features, defectiveness of features, background interference, the complexity of actions, visual content similarities, appearance and pose variations, action's viewpoint change, illumination conditions, etc.

B. Methods of Human Action Recognition in Computer Vision

Recent methods in computer vision for recognizing human actions include first, recognition by depth maps which is developed to recognize the action using depth images [23]. This method is used for capturing motion dynamics of objects from differences of depth images averages.

Second still image-based action recognition, which is based on static information of image only to recognize the action, and only spatial information is provided [24]. This is more challenging than other spatial-temporal based video action analysis for human action recognition due to the limited information is available for the process.

Third, action feature representation methods, for instance, RGB data which includes temporal volume-based (3d spatial-template matching AR), STIP (Spatial-Temporal Interesting Point) which is used to extract key regions of movement in video representation

for AR, Trajectory based features which use paths of tracking key points or joints in human skeleton for representing action [25].

Fourth, depth and skeleton data, which uses motion change of human body depth map and joints of humans between frames of videos [26].

C. Results of Human Action Recognition Tasks in Computer Vision

i. Embedded Computer Vision Application

This stated application was typically developed for intelligence, security, environment, human-computer interactions, etc. The method used includes a combination of Motion History Image (MHI) and Hierarchical Motion History Histogram (HMHH) to represent motion information features and Support Vector Machine for action classification [27].

By applying HHI and HMHH method on the six types of human actions like walking, jogging, running, boxing, hand-clapping, and waving, the author obtained an 80.3 classification rate. But in the task of implementing, the author mainly focuses on computational resources rather than improvement in accuracy further even if it superior to what the author compared with it.

Therefore, it needs more work and its applicability to different recognition tasks. Additionally, feature representation from a motion history image is not always produced good results due to the nature of pixel information that may be hard to detect exact information of objects, especially in motions. So, there should be techniques that should be developed to overcome the challenges of accurate recognition of human action.

ii. Companion Robots and Human Interaction

This work is developed for HAR to support interactions of humans to companion robots. It used HOG as a feature extractor and SVM classifier for recognizing the features [28]. The applied method on three databases contains eight human actions that get an accuracy of 88.7, 74.37, and 51.25.

In the developed system, there should be major room for improvements due to some actions in every three classes are misclassified. From color images robots' motion is unstable and the subject seems to be moving but not. Sometimes the robot and subject are moving. In this case, the accuracy of recognition falls inconsiderably. Therefore, an efficient measure should be developed to overcome the problems.

2.3.2 Deep Learning

Deep learning in its very beginning is a type of machine learning which makes a computer filter inputs through the layers to learn how to classify and predict information [29].

Different from the stated computer vision and other concepts of action recognition, deep learning can learn features (multiple) from many layers hierarchically and builds high-level representations of inputs (raw) automatically. This represents feature selection in case of deep learning is fully automatic. For instance, to learn local features, deep learning uses techniques of sharing weight, local perception, convolutional kernel, pooling, etc. This learning of features is applied to parts of the object, sometimes image instead of the whole part.

Different from the other approaches, this method is fast and enough to obtain object information from rich parts. After learning the features and processes, the final recognition is predicted by layers of multiple convolution results [30]. From the convolution results, one of the most popular approaches to this capability was ConvNet (Convolutional Network). Figure 2.2 shows an example of a deep video recognition using deep learning.

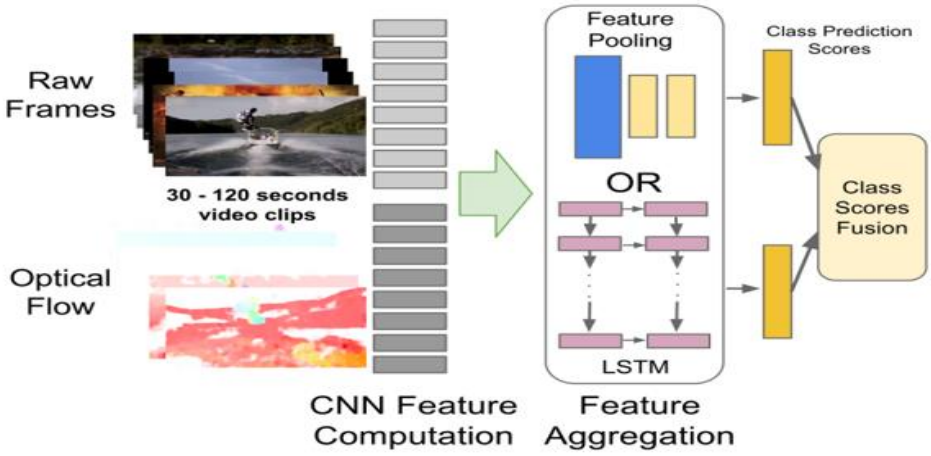


Figure 2.2: Example of deep learning-based video action recognition

Source: Adapted from [31]

As shown in Figure 2.2, multiple features are learned from many frames using convolutional layers, and the learned and selected features can furtherly use for classifications or recognition of the actions. One of the best distinguishability of deep learning from other traditional approaches is the ability to recognize high-level activities with multipart structures.

This is not only the characteristics of deep learning but also the main advantage. Representations of features which are needed to be incorporated by deep learning are STIP (Space-Time Interest Points), local descriptors, body modeling from videos, frequency, RGB video dataset or depth videos, etc.

A. Major Problems of Human Action Recognition in Deep Learning

In recent times, it is obvious that the development of the recurrent neural network, GRNN (Gated Recurrent Neural Network) with increased computational power which is adopted for video classification and sequential data imposes results in many contexts [32].

However, to have an efficient classifier for the class label assignment, there should be a strong feature vector. It is known that feature selection can influence the computational complexity and performance of algorithms [33]. In the case of some deep learning methods, accurate feature selection is not handled rather than common and known techniques used.

The major problems shown in deep learning-based human action recognition were change in viewpoint variations and clutter background, identifying temporal features in a video could be or requires certain sequences of frame and time, video spatial property determinations (pixel locations, grayscale), data processing versus quality of data (Higher quality, longer processing). This is a current video code that can produce very high rate frames with high resolutions (Full HD, UHD), which requires a longer time to process and accuracy vs processing speed in which higher quality can produce better accuracy, but this requires long processing time.

B. Deep Learning Methods and their Approaches for HAR

i. Recognition of Human Action using Sequential Deep Learning

Moez Baccoucheet al. [34] develops deep learning approaches for the recognition of human action which was a sequential technique. This study presents neural-based classifications of human action using deep models that rely on only automatic learning from training examples.

This concept doesn’t use prior modeling the deep networks [34]. It is based on 3-D CNN for automatic learning of spatiotemporal features and then RNN for classifying sequences by considering the temporal evaluation of learned features per each time. Their result shows the application of LSTM models to the system and for taking advantage of automatic learning features, the author evaluated LST with common engineered space-time salient points.

By experimenting on the KTH dataset, the authors get 94.39 and 92.17 on KTH1, KTH2 respectively. However, modeling the system focuses on a two-step process which may increase computational time since the completed model is trained twice. Instead, it is recommended to develop a single step training model to recognize the action which has advantages in reducing the time of computations.

Additionally, since their approach is learning-based feature selection which differs from the handcrafted feature selection approach, the performance evaluation should be on different current challenging datasets that contain realistic scenarios and complex actions. But the authors evaluated their model with only the KTH dataset.

Even if the authors tried to use engineered features of salient space-time, it is based on handcrafted methods that may not be applied for diverse real-world problems and there is no priority-based selection for the frame, i.e. all frames are taken as input for the process of recognizing action. This is one issue in case of decreasing processing time if potential frames are selected from the total video frames. But this idea is not considered in this work.

ii. Model for Human Action Recognition Based on New Hybrid Deep Learning

Neha Jaouedi et al. [35] proposed hybrid deep learning methods for recognizing human action which was a new novel model. The approach consists of two main approaches which include motion tracking by tracking of humans and selection of spatial features. These approaches are developed by first detecting and extracting moving person which is motion tracking by using Gaussian Mixture Modeling (GMM) and Kalman Filter (KF) and second, selection of other features by visual characteristics of each frame on sequences of video using RNN (Recurrent Neural Network) with Gated Recurrent Neural Network.

In the work, the GRU network is used in the node of GRNN which is preferred as lower gates than other RNN models. However, having lower gates may invalidate long term sequences in case of very large video frame sequences. Additionally, in feature selection steps by GMM and KF for motion tracking, here all video frames are equally taken as input to the network. But there may be very similar consecutive frames that may contain similar information. This causes frame redundancy that is one of the consequences of large computation time. Instead, there should be mechanisms by which frame sequences are reduced up to the information of the video that is not altered or diminished. Even in the paper, the authors used GRNN which has GRU at its node for less computation time rather

than other RNN models, the concept of frame priority may also greatly increase time for processing.

iii. Deep Keyframe Detection in Human Action Videos

Such a concept of detecting a keyframe was proposed by Xiang Yan et al. [36]. This is one of the interesting ideas to overcome problems of video action recognitions especially the case of combined factors of background and human pose change in action. It differentiates categories of action from all video action categories as the best contribution in action recognition challenges of video actions.

The Key idea of the task is to generate labeled data automatically for the convolutional neural network by the method of the supervised linear discriminant. So that the model can automatically detect keyframe from action videos of humans. For conducting the task, first labeling each frame as the complete video label, then frame-level label and video for model learning.

For modeling video frame level, assume video as $V = v_1, v_1, \dots, v_k$ where $v_k \in \mathbb{R}^{224*224*3}$ with every frame is represented in RGB and video V belongs to class A, where $A \in \{1,2, \dots, A\}$, then the output of feature map from the convolutional neural network at fully connected layer ($f1, f_2, .., f_k$) and its output from CNN feature map at same fully connected layers of optical flow images (O_1, O_2,.., O_k). These represent optical and RGB features that are concatenated to fuse motion and appearance information of video to apply the capability of representing the video.

For determining the difference between every frame in videos, Linear Discriminant Analysis (LDA) is used. It can discriminate between videos from different classes, reduction of dimensionality, preserving information of discriminatory clauses. To put in practice this idea, aggregated fused features of all videos of the same class and al features of the same class videos are used. Imagine that all features of the same class video $V = V_1, V_2, \dots, V_C$, and all fused features as a matrix of V_C, then from the assumption, class c to be distinguished as class A and all other class B. The comparison will be Class A belongs to class 1, then the feature matrix of class A and B can be:

$$\mathbf{VB} = (\mathbf{V2}, \mathbf{V3}, \dots, \mathbf{VC}) \qquad 2.1$$

The to get projection vector (W_A), we compute Linear Discriminant Analysis as:

$$WA = \text{LDA}(VA, VB) \quad 2.2$$

Finally, compute label value (each frame score) for each class A video training as:

$$F_i, \mathrm{m} = \| F_i, m - W_A{}^T F_i, m_i \|^2 \quad 2.3$$

Where $\mathbf{F_i}$ is an $\mathbf{i^{th}}$ frame feature vector of $\mathbf{m^{th}}$ class A video.

For training the model, RGB video and optical flow images were applied for the learning selection of keyframe models.

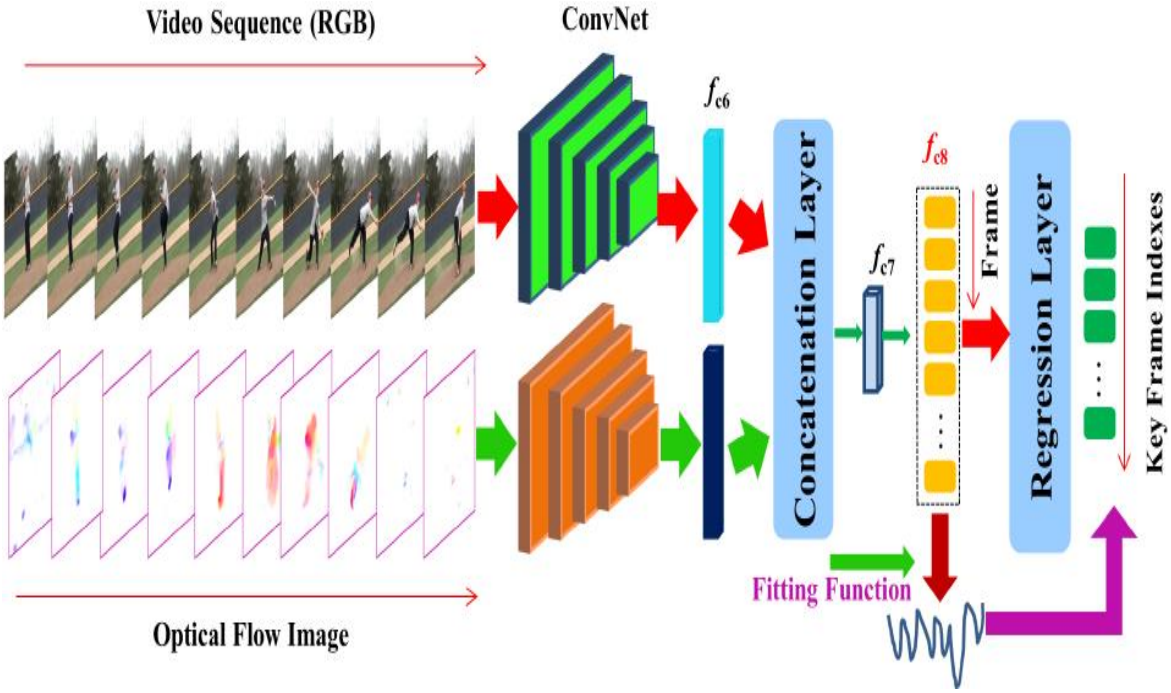


Figure 2.3: Two Stream Conv-Net Keyframe detection framework

Source: Adapted from [37]

From Figure 2.3, there were two networks used, an appearance that works on RGB and motion for optical flow. The appearance-based stream convolutional network is used for the selection of appearance information of f_{c6} layer output and motion stream convolutional network for extracting information of motion of fc layer output at the lower part.

Then feature maps of this network are aggregated to form representations of spatial-temporal features of f_{c7} layer. Then f_{c7} is fed to f_{c8} to form one value feature vector which feeds into to regression layer for optimization of the predicted score. Finally, smooth fitting is applied for all whole videos of predicted scores to distinguish keyframe from whole sequences of videos.

This developed framework is implemented by training the VGG-16 network on the ImageNet dataset from extracting input video motion and appearance features and their optical flow image. Here, selection of video feature frames and optical flow images (f_{c7}) and concatenation into a dimensional visual vector (8192 from Figure 2.3), such visual vector is used for labeling individual frames in a video.

For implementation, RGB or Optical flow images are of size 256x340 and centrally cropped 224x224 and subtracted mean for training. The used dataset is UCF 101 with only 20 classes for training and are used for testing. Since there is no benchmark for keyframe detection, it was not possible to represent the compared results with other methods. Instead, their evaluation criteria used were such as:

***Key Frame Match Error (E_n)*:** Used for detection accuracy evaluation. It is simply the difference between the assumed number of ground truth keyframes in videos (G_n) and their prediction (P_n). Mathematically:

$$\boldsymbol{Keyframe\ Match\ Error(En)} = \mathbf{Pn} - \mathbf{Gn} \qquad 2.4$$

***Key Frame Location Match Error (L)*:** is used for determining exactly if the detected frame is keyframe by temporal location matching.

Mathematically:

$$\boldsymbol{Keyframe\ Location\ Match\ Error(E1)} = \pm \sum_{x=1}^{k} |\mathbf{Pl(x)} - \mathbf{Gl(x)}| \qquad 2.5$$

Where $\mathbf{P_l^{(x)}}$ is the predicted keyframe location and $\mathbf{G_l^{(x)}}|$ is ground truth keyframe location, **l** is the location of the keyframe. By applying the method on a UCF101 data set, the authors got locations and keyframe matching accuracy 77.3. To this work, the authors reported their proposed work as the first keyframe detection in human action recognition. To represent its strength, it provides the base for overcoming challenges in labeling frames in a video in case there are no frame-level labels for instance UCF101 datasets.

The proposed method can automatically generate frame-level labels by combining Linear Discriminant Analysis and Convolutional Neural Network features of frames. Even if it is

one good approach to detecting action from videos, there is another room of problem which someone should consider. This issue is, what if only relevant features of selected keyframes are learned in the network for improving performance, time of computations, and storage, or even other resources.

This approach is not studied by the author of the mentioned work. For the word implanted which entitled "Key Frame-Based Saliency Detection for Human Action Recognition Using Deep Learning", the stated problem is considered and solved by developing a saliency detection model for keyframe selection in human action recognition system.

2.4 Major Application Areas of Key Frame Saliency Detection

2.4.1 Salient Object Detection

Object detection is mostly attracted computer vision and its approaches and in recent times the idea of saliency detection gets much attention. This result is much shown in the medical diagnosis system where the infected tissue is clearly shown separate from other parts and this facilitates the capability to identify the disease for ease of treatment. For instance, brain tumor, cancer, etc. require high accuracy to detect infected region and treat easily. But, in ancient times, it is difficult to detect the correct location of the infected region and this needs surgical activity which may hold total part or half surging to understand the disease.

Source[1]

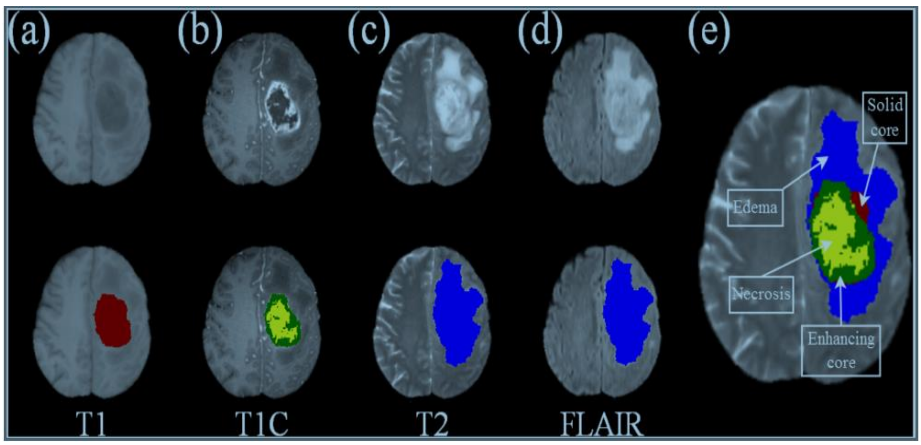


Figure 2.4: Intratumor structures

Figure 2.4 shows the tumor disease detection by applying saliency detection for separating areas affected by the disease from an uninfected region. Even for more visualization

[1] https://journals.plos.org/plosone/article/figure?id=10.1371/journal.pone.0187209.g001

purposes, multi-color detection was applied. Additionally, deep learning-based saliency object detection is used due to its capability to extract high-level features and details of low-level feature information. E.g. CNN. From the applicability of deep learning for salient object detection, it possible to exactly locating a salient object place that can be done. For instance, the use of the Global Guiding Module for providing information to layers at different levels locating the potential salient object and Feature Aggregation module, for fusing coarse level and fine level features. This idea is shown in Figure 2.5.

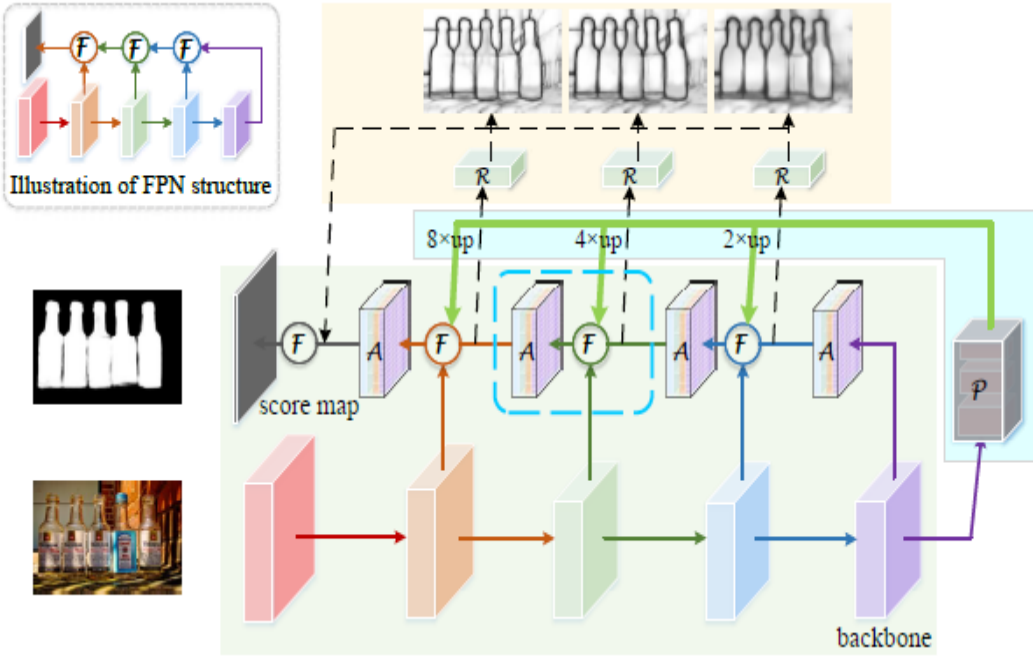


Figure 2.5: Pooling based salient object detection

Source: Adapted from [38]

2.4.2 Human Activity Recognition

Human Activity Recognition is the process of representing and recognizing person actions from the observation of a person and its environment. This needs accurately locating movements person and appearance concerning scenes where the activities are proceeds. Among recognition techniques, deep learning gets much attention due to its nature of understanding and learning multiple features in multiple layers. Here recognition and accuracy of the method may also improve the system capability to overcome different challenges in the process. Figure 2.6 shows human action recognition using saliency estimation. The estimation includes temporal saliency for frames and salience regions. The

bot the estimated temporal and salience regions were extracted using HOG. Finally, downsampling the extracted saliency based on time and recognize the action.

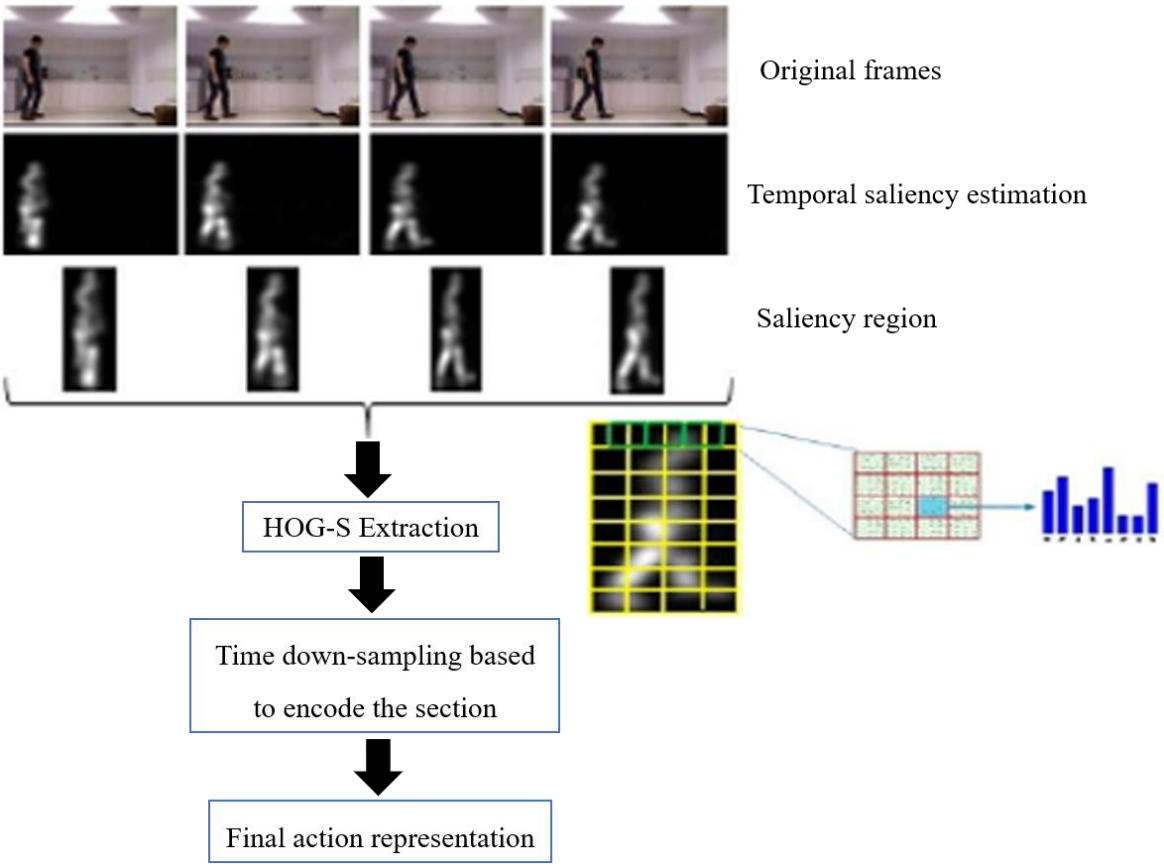


Figure 2.6: Saliency Based Human Action Recognition

Source: Adapted from [39]

2.4.3 Video Keyframe Extraction

This is the process of mining representative potential frames from video clips. The selected frames can have the ability to represent other frames in the same clip and it contains potential information which mostly rich information than others. In the case of detecting and extracting video frames, the process is challenging due to different factors related to the poses of human beings and dynamic background changes. But its crucial application is video interpretation, summarization, indexing, compression, etc. The concept of extracting keyframes from video used was two-stream ConvNet for capturing the location of keyframes in the videos and distinguish keyframes from other non-keyframe sequences in a video. To extract the keyframes, deep features of frame sequences were captured and learned. Figure 2.7 shows extracting keyframes from a deep video.

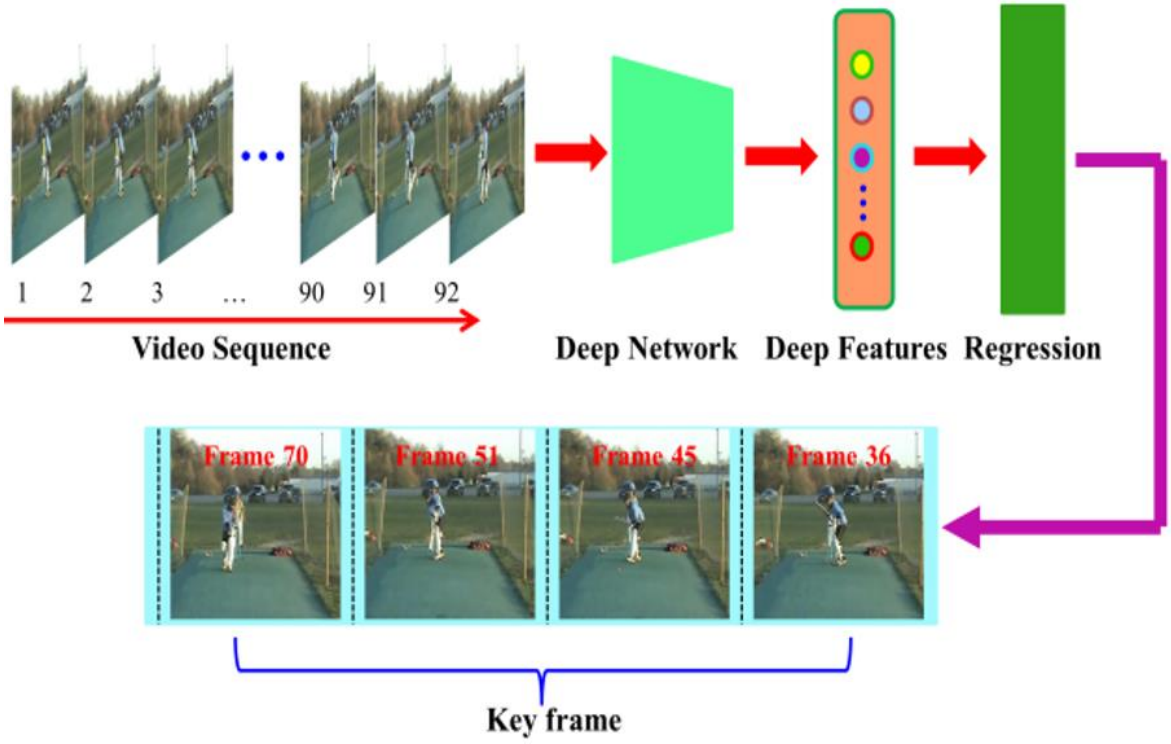


Figure 2.7: Deep Keyframe Detection in Human Action

Source: Adapted from [40]

Figure 2.7 shows the detection of deep keyframes in human action video. From the sequences of frames in a video, keyframes were detected using two-stream ConvNet which learns the location of keyframes. The network contains two important frameworks, summarizer, and discriminator. Summarizer captures the appearance and motion of video frame features and encodes them for representing the video. Finally, the discriminator is used as a fitting function for distinguishing the delectated keyframes and other frames in the video.

2.5 Related Works

Theirry Baumann .et al [41], proposed the concept of background subtraction in the real application, its challenges, current models, and future directions. Their objective was to study real applications that use the background subtraction method and then provide exhaustive information on challenges real practice as well as providing future direction. Their study focuses on areas of applications such as surveillance (traffic, airport, maritime), Human activities (sports, athletic performance, surveillance in dangerous activities), and human-computer interactions (game, ludo-applications).

By studying the area, the identified problems such as camera jittering, illumination change, dynamic background change of outdoor scene. As a future direction, the authors recommend using recent background subtraction methods (fuzzy models, RPCA, deep learning), developing specific background models for a specific application or universal background model for all applications.

However, their study is only focused on theoretical study, there is nothing supported with experimental tests, results. Even, the authors didn't consider its implementation with most advanced algorithms like deep learning, but the focused-on machine learning approach and computer vision approaches. This limitation was solved in the proposed work by experimenting with complex scene background and foreground data to be used in implementation versus the separated foreground from its background which was performed by the saliency detection method.

Chen. et al. [42], developed saliency using context two-stream ConvNet for action recognition. Their work aims to recognize the entire scene in video frames for action recognition by developing and employing two-stream ConvNet.

The authors used camera motion estimation and salient human motion as a method. By applying their method on the UCF101 dataset, the author gets 92.4% accuracy. But the author didn't focus on the concept of frame variation which exists in real-world problems. This challenge is solved in the proposed work by estimating temporal frame segmentation and keyframe selection approach to limit the problem of frame replication that can cause redundancy and lots of time to compute.

Lamin Wang .et al. [10], develops Temporal Segment Network for recognizing action from videos. The objective of the work was to develop a long-range temporal structure through a new aggregation module and segment-based sampling for action recognition from video. Based on the proposed method the authors suggested that was an efficient and effective method to capture long-range temporal structure and possibility to train on a limited training set without simple overfitting.

In their work, the focused-on segmenting temporal video was based on the nature of the network and developed optical flow for each selected short snippet from the frame. But, choosing the snippet from the frame is based on the temporal sliding window of different size approach of frame score is used in which maximum window score represent it and the whole video prediction after aggregation of different window size. Beyond their concept,

optical dependency motion information may be challenging in the complex background with fast motion.

Therefore, the case of selecting snippets from the segmented video frame is solved by the proposed work which is conducted by computing variation between fame sequences and selection of the frame by the Key Frame selection technique.

Xiang Yan.et al. [36], developed deep keyframe detection in human action videos. Work aims to detect typical frames in videos using human action. The authors applied two-stream ConvNet (summarizer and discriminant) for predicting directly keyframe locations. By experimenting on the UCF101 dataset of 20 class ground-truth the authors get an average accuracy of 0.773 location for matching 2.02.

Even if the authors posted as it is the first work done on two-stream ConvNet and nothing about its weakness rather than the non-existence of the benchmark dataset, the author did not consider how about extracting the most relevant features from the scene, but they prefer detecting keyframe only. For the work implemented, it is better if we focus on the most efficient and usable feature when selecting keyframes from an object of interest.

So, the efficiency of the process may be achieved in this way. In the feature, this challenge is solved by the proposed work by using salient detection which focused on most visualized and relevant parts of objects instead of the total object scene fed to the system.

Lai Jaing .et al. [43], proposed a video saliency prediction approach based on deep learning (i.e. Deep VS) and Saliency Structured Conv-LSTM network using features selected from Object Motion CNN. Here OM-CNN is used for predicting intraframe saliency and SS Conv-LSTM is for modeling interframe saliency of videos.

The authors were used cross-net and hierarchical normalization to combine temporal and spatial features of motion and object-ness subnet. An Interframe saliency map is generated which considers the cross-frame transition map of human attention map and structured output with center bias by using SS-Conv LSTM. By applying the method on their dataset, the author got mean (SD) saliency prediction accuracy (AUC=0.90) and 0.81, 0.86 over SFU, DIEM dataset respectively.

According to their implementation, their method outperforms public eye tracking databases. But still, their consideration of video frames is inclusive of each frame. The problem needed to be specified is what if there are nearly or redundant frames that may be required to be

eliminated from the sequence for fast processing. This question was not addressed in their work. Therefore, such a problem was solved with consideration of selecting keyframes from video sequences by measuring the difference between keyframes and computing the local maximum to be a keyframe and non-key frame identification in the proposed work.

Ashwan Anwer [44] proposed human action recognition by using saliency-based local and global features. 3D SIFT, HOOF local, and global descriptors respectively are used with a multi-class SVM classifier for HAR. Moreover, the Gradient Location and Orientation histogram which extension of 3D is used for local feature representation of both their relative location and gradient orientation. BOW is used for mapping video sequence's local features to the pre-learned dictionary developed by the K-means algorithm. Matching skeleton graph combined with pairwise graph similarity using optimal subsequence bijection with dynamic wrapping time used to handle temporal and topological variations robustly.

This work [44], however, depends on the selection of frame should be consistent, common, periodic, and starting frame. The approach is based on computer vision and most challenges remain to exist in the research. For instance, SIFT is based on the computation of gradients of each pixel in the patch of the histogram of gradients which is complicated and heavy.

Additionally, it can include noises into features if not carefully undertaken and may lose spatial information. In the work, however, the process of selecting the keyframe was not valid that always starting frame should represent the action. It may be the first, last, or intermediate frame that needs a deep understanding frame's potential information. Additionally, if the action is not from common and periodic, the network may fail to correctly recognize the type of action.

Therefore, such challenges are further studied in the proposed research by using a deep learning approach for human action recognition. In the case of this study, reviewing how to represent action from video frames by using only selected frames that have potential and only relevant information is selected from the selected frame for further process is conducted.

Zengmin et al.[45] develops recognition of action by using saliency-based dense sampling. This task was carried out by using an improved dense trajectory (IDT) on a salient region-based contrast boundary (IDT-RCB). The basic idea is to automatically estimate the region of salient objects across each frame and improving the IDT sampling method without knowledge of video content. For better recognition results the author fused it with CNN.

However, their approach is based on handcrafted features that may lack the discriminative capability to represent the action. Moreover, in their work, the authors ignored static trajectories and those large displacements due to inaccurate optical flow that makes incorrect results. But the problem was, what if the removed or ignored trajectories have the potential to represent the action? This challenge was not addressed in their work.

Additionally, region-based contrast mapping may be inaccurate if the object has the same foreground and background color that may exist in a real-world problem. Even if the authors divided it into different colors by using grapy cut and develop a histogram for each region, they may have a relatively similar result. Also, optical flow can be challenging for fast motion and high illumination changes such as in outdoor environmental conditions.

Therefore, in this proposed study deep learning-based approach to recognize the action is preferred. This is due to the deep learning method to automatically learn features and semantic representations from raw videos by using deep neural networks.

Shichao et al.[46] proposed a static history image (SHI) and sub-action motion history (SMHI) based action recognition. Their basic idea is improving action recognition using depth maps. It is based on portioning video into different sub-action segments and computing the Local Binary Point (LBP) descriptor for representing an action.

The author used principal component analysis (PCA) for feature selection and support vector machine (SVM) for classification. By experimenting on the MSR 3D dataset the authors achieved accuracy 93.7%. However, the algorithm works well if the intra-class variation is low.

Additionally, it is challenging while multiple actions were needed to be undertaken for classification. Moreover, it is not always valid for representing human motion information using SMHI, since exact motion cues detected by the algorithm may be complex. Although the good performance was obtained, it is challenging to apply for real-world problems because the selected algorithm works well for similar action categories which are based on low intraclass variations of action categories.

Instead, in the proposed study real-world and challenging action categories were undertaken to recognize the action types. Additionally, multiple action categories were studied in the proposed work. A deep learning-based method is applied to overcome challenges of

dependability on a few action categories to get better accuracy instead of only classical computer vision algorithms.

Andargie Mekonnen et al.[47] proposed an optical flow-based classification of Ethiopian traditional dance video. The author used the magnitude of optical flow to reduce training complexity and real-time content-based classification of video stability.

Motion features are selected using optical flow and Artificial Neural Network is used for action classification. By applying on the collected dataset, theys get overall accuracy 80%. However, challenges such as inter similarity between selected features by two optical flow algorithms made low accuracy. Additionally, the correlation between dance style remains challenging. Moreover, action recognition from the optical flow of the dancer made the chance of misclassification due to factors such as clothes, women's hairstyles, and objects.

Therefore, the achieved overall accuracy is not efficient for real-time applications and even optical flow-based method concepts such as scale and direction should be considered instead of only depending on the magnitude. For this reason, in the proposed study, how to include frame sequences and selecting features for recognizing action is done with further contributions.

Dr. KarpagaSelvi Subramanian and Andargie Mekonnen [48] proposed a content-based classification of Ethiopian traditional dance using HOG and Optical Flow. The aim of their proposed work was classification of traditional Ethiopian dances by identifying gestures and movements specific to traditional dances.

The author used HOG for detecting region of interest to locate humans in a sequence of time-ordered images. Similarly, they used Optical Flow for detecting motion and extracting feature vectors from the sequences. SVM is used for classifying the actions selected by Optical Flow as feature vectors.

Using selected traditional video (100) of ten (10) classes from the entertainment website the authors achieved an overall accuracy of 78.1%, 64.6%, 99.7%, and 98% on Lucas-Kanade, Horn-Schunck, HOG of Lucas-Kanade, and HOG of Hom-Schunck respectively. However, problems such as human-based annotation, used data fit to proposed scope, action classification problems such as dynamic background change, camera motion, visual similarity, and appearance variation, inter or intraclass similarity of actions and nature of

actors performing similar activities in a different style or vice versa was still a big gap to be fulfilled.

As the author stated, the optical flow was used for computing consecutive frame motion differences. But, if the optical flow is preferred and chosen by such implementation, it should be applied to video data rather than selected consecutive frames for better results. Moreover, the prototype was not developed in their proposed work.

To overcome such problems raised in the work, we proposed a deep learning approach while classifying actions and challenges related to data processing were solved using selecting potential frames from each video and detecting their salient information and using only relevant features using saliency detection methods. Finally, informative and relevant sequences of frames were given to classifiers for efficiently classifying the action. A sample of related papers reviewed in this work was described in Table 2.1, at the end of this chapter.

2.6 Summary

The purpose of this review is to identify major studies related to human action recognition, their problems, the gap to be filled, and the method used in the system. Through the study, it is possible to understand that the existing system uses a method different from currently developed in this study. This means the existing action recognition uses inclusive (whole) video frames, even if the authors used selected keyframe, lack of detection only potential features from the object action still existed. This resulted in resource consumption such as time, storage, and, etc. Therefore, applying a new method that uses only informative frames from video sequences and selecting only necessary features from the frame to recognize the action is preferred.

Table 2.1: Summary of Sample of Related Papers

No	***Title***	***Objectives***	***Methodology***	***Limitation***
1	Saliency- context two-stream convnet for action recognition /Chen .et al [15]	Studying real applications that use background subtraction.	Saliency and Context ConvNet	Non focused frame variation approach
2	Deep keyframe detection in human action videos/ Xiang Yan.et al [16]	Detecting typical frames in videos using human action	Two stream ConvNet: (Summarizer, Discriminant)	Lacks selection of interest features
3	Video saliency prediction approach based on DL/Lai Jaing et al. [17]	Developing DL based video saliency prediction	Deep VS: OMNN (Spatial-Temporal intra-frame SV and SS-ConvLSTM (interframe SV)	Inclusiveness of all frames without giving priority.
4	Background subtraction in a real application, its challenges, current models and future directions/ Theirry Baumann .et al [14]	Studying real applications that use the background subtraction method and provide info.	Theoretical study on different application areas of background subtraction	Nothing stated implementation results rather theoretical only
5	Human Action Recognition Based on Foreground Trajectory and Motion Difference Descriptors/ Suge Dong et al. [49]	To overcome high redundancy trajectory and susceptibility to background interference HAR	Foreground region extraction by dense optical field, MDD for relative motion info, fisher vector to encode features, and SVM	Local minima point drifting in extracting foreground b/c of moving point traversing

6	Action Recognition in Video Sequences using Deep Bi-directional LSTM with CNN Features	Developing action recognition method by CNN and deep Bi- LSTM	CNN for deep feature extraction, DB-LSTM for sequence learning, SoftMax for classification	The whole frame used and non-analyzed only frame salient regions
7	Temporal Segment Network for recognizing action from videos/ Temporal Segment Network for recognizing action from videos, 2019 [10]	to develop a long-range temporal structure through a new aggregation module and segment-based sampling for action recognition from video	Frame segment using RGB difference, RGB frames, optical flows, temporal convnet and spatial convenet aggregation and finally class score fusion for segmental consus	Short snipnet may not contain all the representative frames. Optical flow for motion detection can cause discontinuity for multi action and fail for uniform action.
8	Human action recognition by using saliency-based local and global features/ Ashwan Anwer/2017 [44]	Human action recognition using saliency-based global and local future representation	GLOH for local feature extraction, BOW for mapping of video sequence local features, 3D SIFT for local feature , HOOF for global features and SVM for classification	Depends on consistency of frame selection, common and periodic frame. SIFT based feature representation is complicated and heavy process.

CHAPTER THREE

3. RESEARCH METHODOLOGY

3.1 Introduction

This chapter describes the way by which keyframe saliency detection was undertaken including the method used. This includes ways of collecting, processing, tools, or materials used in collecting data, method selection, and ethical considerations.

Additionally, it covers systematic methods, practices, and approaches to conduct deep learning-based human action recognition based on keyframe saliency detection. This includes ways of collecting, processing, tools, or materials used in collecting data and selecting the method.

Therefore, techniques used to achieve the stated procedure includes:

Literature Review: studying different sources such as papers, magazines, journal articles, workshops, tutorial, etc. for understanding current human action recognition regarding existing problems, the method used to overcome the problems, the gap to be fulfilled, weakness and strength of existing work done on it and developing a new method to fill the gap and identified problems.

Data Collection: selecting 20 classes from UCF101 Dataset, [57], which is available from the UCF repository and processing for use was done in this work .

Developing KFBSD Model: to recognize action using the proposed keyframe based saliency detection, its model is developed and used while recognizing the action using VGG-16 + bi-directional LSTM model. To a developed and implement model, setting environmental using python, OpenCV, TensorFlow backend as Keras were undertaken.

Evaluation: after developing the model, implementing for recognizing the action, evaluating the model performance using accuracy and loss at both training and validation, comparing the developed model with existing human action recognition models such as CNN, VGG-LSTM and VGG-Bi-LSTM were performed. The model evaluation result was described using accuracy, validation and the assessment of model was represented by confusion matrix.

3.2 Data Collection

3.2.1 Data Collection Method

To accomplish this study, a freely online available dataset was selected and employed due to its availability, accessibility, and efficiency to employ. Additionally, since the proposed method is almost generic, the available online dataset regarding human action recognition is preferred. The selection of datasets is made with considerations such as the relevance of data to the area of study, nature, and quality of data. The stated issue was considered due to the existing human action dataset such as YouTube, Sport 152, Kinetics were huge and in some cases for ease of manageability to the developed model with the required time [57]. The type of tools or materials used in collecting these types of data was HD video cameras with high quality, smartphone, and very high-quality sensors.

3.2.2 Description of the Data

In this study, all data used were video action which is collected and used as pre-input to the developed system. The data were already collected was from different areas of available data, such as from YouTube, Sports field, and stored as a data set in the repository of UCF101 (University of Central Florida) which are freely available online.

UCF101 dataset is one of the action recognition datasets which has 101 action categories and these categories were divided into 5 types [57]. UCF50 dataset has 50 action categories. UCF-101 comprised 13,320 videos and all video actions were grouped into 25 in which each group contains 4 up to 7 videos. It aims to inspire more analysis for recognition of action by exploring and learning new realistic categories of actions. The five types of this dataset categories were human-object interactions, human-human interaction, body motion only, Playing music instruments, and sports. Some of the action categories of the UCF101 dataset includes: Apply Lipstick, Billard, Bowling, JanpingJack, HauloHoop,PlayingDhol, PlayingGuitar, IceDancing, Kayaking, etc. [57].

Generally, for timely processing and using accessible resource constraints, only 20 class among 101 class of UCF-101 datasets were selected to train and validate the developed KFSD HAR model. These data which are used in this work were from action types including 7 from Human-Object interaction, 9 from sport, 3 from playing musical instruments, and 1 from body motion only. All the selected classes were first processed using keyframe selection to select potential frames used to represent the action later. Second, the selected keyframe was detected using a saliency detection network to include only relevant features

per frame. Finally, reconstructed videos were fed to VGG-16 + Bi-directional LSTM to recognize the action. To implement the model, processed data were divided into training and testing set split using a machine learning approach. Therefore, in this study, a 70% training set and 30% test set were used for training and validating the data on the developed model. On the otherhand, the same procedure was applied for predicting the model except the total number of data used were randomly selected 5 classes of action videos containing 500 video.

3.3 Data Processing and Procedures

In this stage, accessing the data from the source (UCF) and selecting the datasets into proper directories is done as the first step. By its nature UCF101 is prepared with considerations of categories and types of actions performed in each category. Second, all the accessed data is selected to the specified directory, size, and type of each action were checked. Finally, the relevance of actions to the action category was evaluated.

Even if the computerized system was used for processing the data, these steps take time since the prepared data is almost usable regardless of their sizes and resolutions.To simplify the tasks of data processing, series of steps were followed which includes: collecting human action recognition data from the UCF101 dataset (Collection), storing the collected dataset in digital form in the specified directory (Storage), and selecting the most relevant action categories and storing in separate storage (sorting)

Data preprocessing employed in this study contains two main categories, namely (1) preprocessing for keyframe selection, which includes: arranging video file formats (.AVI), resizing video data (324x224), segmenting video clips into equal frame sequences, and storing in JPG format, RGB differencing and computing the local maximum of segmented video frames and (2) preprocessing for saliency detection, which includes: resizing images (320 x 240), changing the color image to grayscale (0,1) and computing image saliency (image saliency map, attention, edge detection).

3.3.1 Programming Tools Used

In this study, tools used for making it easy to sort data and identify relationships, patterns, trends, correlations, and anomalies that may otherwise hard to detect were selected. Additionally, this tool makes it easy to manipulate and process data, analyze the relationship and correlation between datasets.

In this proposed work the preferred programming tool for conducting the task was Python. It is an interactive programming language integrated with other languages including, Java, PHP, HTML, CSS, etc. In this study, the integrated package with python was OpenCV which is used as open-source libraries and common infrastructures provide for applications of computer vision and accelerating the use of machine perception.

Additionally, it was used to integrate different machine learning libraries and deep learning frameworks. Moreover, a deep learning framework, Keras backend TensorFlow**,** allows different machine learning and other deep learning libraries to read different syntax and provides excellent functionalities. So, in this study, Keras backend TensorFlow was used as a deep learning framework.

Generally, for implementing the proposed keyframe-based saliency detection for human action recognition, python programming tool and deep learning framework Keras backend TensorFlow and libraries such as OpenCV, Pandas, etc. were used. The summary of tools such as hardware, software, framework, and languages used in this proposed keyframe-based saliency detection for human action recognition was given in Table 3.1.

Table 3.1: Experiment Tools used for KFSD HAR

S.No	***Experimental Setup***		***Description***
1	**Hardware**	Computer	Desktop: Dell OptiPlex-9020
		Processor	Gigabyte RTX TITAN-2070 (GPU)
			Intel(R) Core (TM) i5-7200U CPU
		Memory	8GB RAM
		Power Supply	ZOTAC 620W
2	**Software**	Anaconda Python	Version: 3.7.4
		Cuda Toolkit	Version: 10.07.63
		CudCNN	Version: 10.0
3	**Language**	Python	Version: 3.7.4
4	**Framework**	Keras backend Tensorflow	TensorFlow-GPU backend with Keras
5	**Libraries**	OpenCv, Panda, MatplotLib, etc.	Opencv 3.4.5, Panda and Matplot Library

3.4 Method Selection

As stated, action recognition from a video is quite challenging due to consistency and dynamic background or motion of object changes. Additionally, during extracting features, the existing system does not much consider only relevant parts of selected frames. This causes a huge storage requirement and longtime computation. But in this study, there was a contribution made with considerable problems of current human action recognitions.

Therefore, in this study keyframe-based saliency detection method solves the problem of including feature selection using only relevant features by contributing major roles in the tasks of human action recognition. This employs roles such as the selection of keyframes from video action categories which extracts frames those rich in potential information to represent the type of action using keyframe selection algorithm and detects salient regions of keyframes selected to include only actions relevant information using saliency detection for making ease of extracting feature in the action recognition stage. By doing so, the process of human action recognition becomes an easy task.

3.5 Designing Keyframe Based Saliency Detection for HAR

3.5.1 Designing Tool

For designing key frame-based saliency detection architecture or model, process flow in the network for human action recognition, MS-Visio 2019 tool was used. It is a diagramming tool that contains basic UML diagrams, charts, schedule, database drawings, network diagrams, maps and floor plans, software, and general drawing. Additionally, MS-Visio is simple and integrated tools with Microsoft office products which are easy for drawing diagrams and needed models.

Since MS-Visio is integrated with MS-Office for instance, Office Word 2016 or 2019, anyone can access from his or her installed office word and use effectively. This reduces the need of having products from separate areas and working environments. So that at the same time working on office word, anyone can access MS-Visio from the same environment.

Even though there was much software used for drawing and designing a system like an enterprise architecture, which is common software for designing UML software diagrams, charts, and symbols, Edraw-Max, almost such software has similar functionality for drawing compared to MS-Visio. Their difficulty is the need for license agreement and even need knowledge for use.

Generally, the reason for selecting MS-Visio was due to ease of understanding, less knowledge required for use, ease of drawing, simple to integrate with Office-Word, precise and clear drawing which doesn't need reformatting or editing while using in other systems, simple to access, license-free product. The concept of frame segmentation was adapted from [10]. Figure 3.1 represents the design of keyframe based saliency detection for human action recognition.

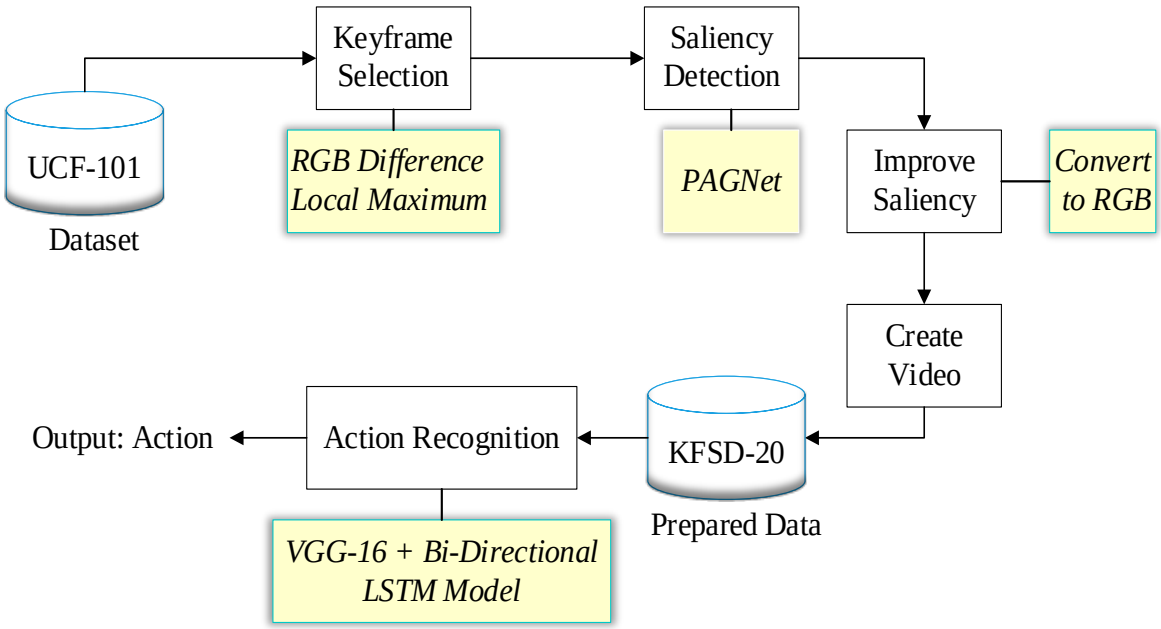


Figure 3.1: Keyframe-Based Saliency Detection Design for HAR

As shown in Figure 3.1 in the above, the process of designing the proposed KFSD HAR starts with selecting UCF-101 datasets with 20 classes of action categories and applying the KFSD method. From videos of 20 action categories, keyframes were selected using the keyframe selection algorithm and the selected keyframe was detection using the saliency detection method.

The result obtained from the keyframe-based saliency detection method was used for the action recognition network after converted to video. Generally, the designed KFSD HAR includes data, keyframe-based saliency detection, and action recognition networks.

3.6 Experimentation

For experiments on the proposed system, environments and experimental setups were adjusted accordingly for the success of the tasks. In this study hardware, software, languages, and framework used were described. The descriptions of these setups were given in Table 3.1.

The conducted experiment includes, video was converted to frame sequences, keyframes were selected from frame sequences using keyframe selection method, keyframes were detected using saliency detection model, Pyramidal Guided Attention Network, then salient frames were converted to video and fed to action recognition network which is Bidirectional LSTM. In this network, first features were extracted using VGG-16 and frame sequences were learned using forward and backward layers of LSTM. Finally, action were classified using SoftMax classifier.

3.7 Performance Evaluation

In this study, the performance of the developed system was conducted by applying evaluation metrics using accuracy and loss.

Accuracy: according to [50], accuracy is expressed as the percentage of queries that the model can be able to predict the correct label.

According to [51], True positive (TP) is an outcome where the model correctly predicts positive classes. Whereas True Negative (TN) is a result obtained when the model correctly predicts negative classes. Similarly, False Negative (FP) is when the model incorrectly predicts positive classes and False Negative (FN) is the outcome obtained where the model incorrectly predicts negative classes.

Suppose that a classification model to classify data into two classes namely actual and predicted. The class can be actual positive class predicted as positive (TP), actual negative class predicted as negative (TN), actual negative class predicted as positive (FP) and actual positive class predicted as negative (FN) then accuracy can be defined as:

$$\boldsymbol{Accuracy} = \frac{\text{No of correct predictions}}{Total\ No\ of\ predictions} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad 3.1$$

Catagorical Cross-entropy loss: used for computing the loss of multi-classification action used in this study with encoded using on-hot approach. This type of loss computation is preferred because it measures how two discreate probability distributions are distnigishable from each other. So, the probability of an event i occur in y_i and the sum of all y_i =1, meaning exactly one event may occur.

$$\boldsymbol{CCE} = -\sum_{i=0}^{n} y_i * log(\acute{y}_i) \qquad 3.2$$

Where, CCE is categorical cross entropy, y_i is target value and ýi the output value and n is number of output size starting from i = 0. The minus sign represents the computed loss get smaller while distribution gets close to each other.

For assessing the the performane model, a confusion matrix was used. It is a clear representation of prediction results. It is a summarized table of assessed classification model performance which is also known as error matrix. By using the confusion matrix number of incorrect and correct predictions was given as summarized with count values that are broken down using each class. By definition confusion matrix C_{ij} is the number of observations made known to be in a group *i* and predicted to be in group *j*. It visualizes the accuracy of the classifier by comparing the actual class with the predicted one.

3.8 Summary

This chapter described the purpose of developing a method to conceptualize, ways of developing and implementing the proposed methodology, understanding the basics and nature of data, process, analysis, tools used, and ways of assessing experiments to be conducted. Moreover, the reason for choosing concepts behind some points is described with their evidence to their possibility, accessibility, and validity cases. In a general way, it represents how to conduct and put into work the proposed study based on the explained methodology.

CHAPTER FOUR

4. PROPOSED KEYFRAME-BASED SALIENCY DETECTION APPROACH

4.1 Introduction

This chapter describes the model, algorithms used in the implementation of the keyframe-based saliency detection for the human action recognition model, basic process, and steps used to develop the model. Additionally, the result obtained by each algorithm of the model was given under each section. Since their nature guides the process, stepwise implementation is preferred. For instance, keyframe selection and saliency detection as a sequential process in one category, and action recognition in the next step. The process flow in model is shown in Figure 4.1,

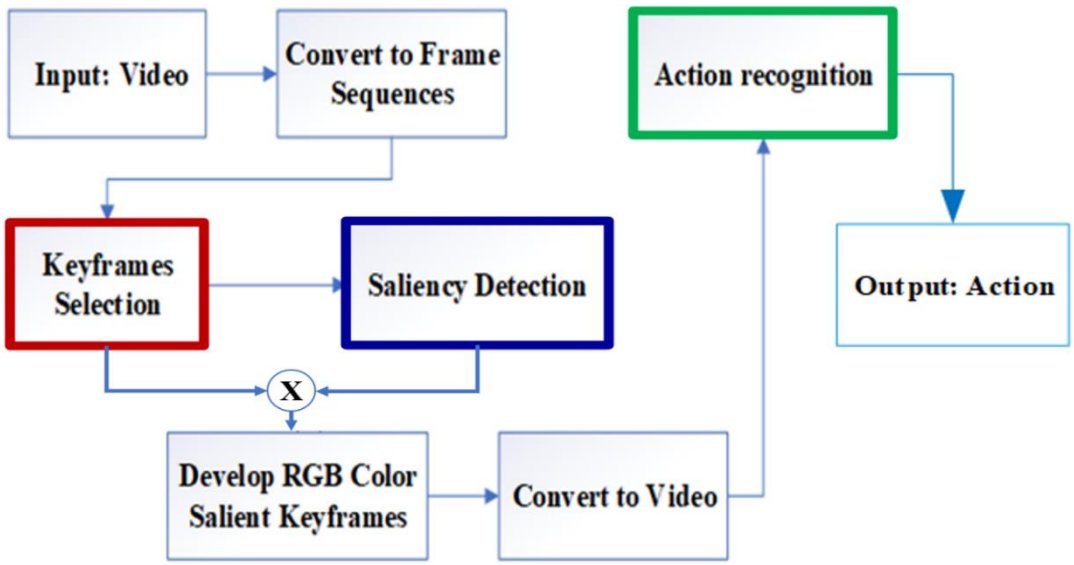


Figure 4.1: KFSD HAR Model Process Flow

As shown in Figure 4.1, keyframe selection algorithm selects potential and informative frame from video action. Then using selected keyframe, saliency detection was applied for detecting salient regions of actions. Finally, a reconverted video of the salient keyframe was given to the action recognition network to recognize the action. Even if the proposed work contains three algorithms, initially keyframe selection and saliency detection are linked in end-to-end, and then finally, the action recognition network was presented hereunder. The general overview of model was given in Figure 4.2.

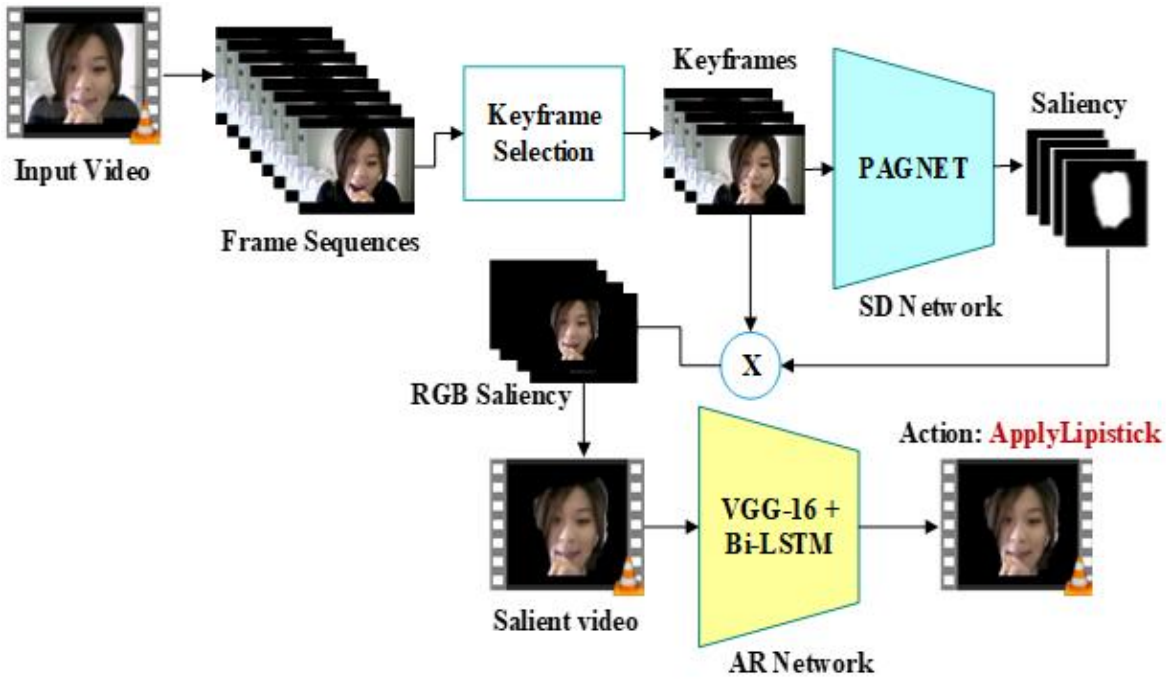


Figure 4.2: General KFSD HAR Model

4.2 Key Frame Selection

Keyframe selection from video data is one of the research areas of information retrieval and video object recognition. It is nothing but a frame of image in a video sequence that is representative and contains potential information to represent the whole so that it can reflect the video content summary. By using the keyframe selection, the content of video data can be expressed clearly and the amount of memory needed for processing video data and complexity is reduced. This plays a role in applications of video summarizations, video object recognition, classification, archival surveillance, and retrieval of video.

4.2.1 Problem Formulation

In the current video recognition applications such as surveillance, ambient assistive living, traffic monitoring system, violent activity recognition such as crime detection requires a huge volume of video data. For instance, a video collected at 25fps in 30 min contains 25*1,800 = 45,000 frames. This requires large computation and storage with high processing computers and high processors such as multi GPU.

In this study, to overcome the computation, storage, and high processing problem only potential frames were selected to represent an entire video. The potential keyframe was selected by computing the inter-frame difference of video sequences, where the output of

this stage is an input for the saliency detection. The selected frame contains information to represent the left frames in the same category.

4.2.2 Key Frame Selection Algorithm

The algorithm is based on the approach of inter-frame difference computation for video sequences. This algorithm is described as the following:

Algorithm 1: Keyframe Selection

Input: Video, V

Output: keyframes

***Set window length*:** win_len

Set number of top frames: NUM_TOP_FRAMES

Begin

```
cap = cv2.VideoCapture(str(videopath))//read data
Frame, success = read(V), i = 0
  curr_frame, prev_frame, frame_diffs, frames = [], [],
[], []
  while (success):
     diff = abs_diff(curr_frame, prev_frame), diff_sum
= sum(diff)
     diff_sum_mean = diff_sum/ diff.shape[0] *
diff.shape[1]
     frame_diff.append(diff_sum_mean)
     frame = frame (i, diff_sum_mean), i = i+1
     frame = read(frame), sim_diff_array = smooth
(frame_diffs, len_win)
     frame_indexes =
store.asarray(argrelextrema(sm_diff_array,
np.greater))[0]
  for range (len(in frame_indexes):
     key_frame_id = add(frame[i-1].id), idx = 0
  while (success):
      if idx in range(len(key_frame_id):
      write frame (Keyframe_Path + folder_name[:-4]+
"001_{0:06d}.jpg".format(int(idx))), frame))//sequence
      idx = idx+1
```

End

In the above algorithm, selecting frames from video data is performed by reading the video file and selection of frames by computing the inter-frame difference. This is followed by computing frames similarity. The difference between interframes is smoothed before calculating the local maximum. Then based on the average of local maximum, frame with

greater value can be stored as keyframes and those couldn't fulfill the requirements were discarded.

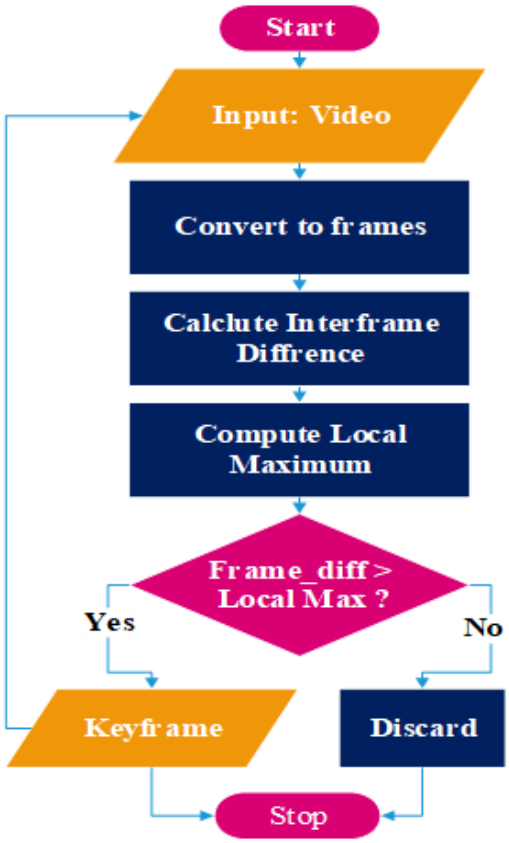


Figure 4.3: Key Frame Selection Process

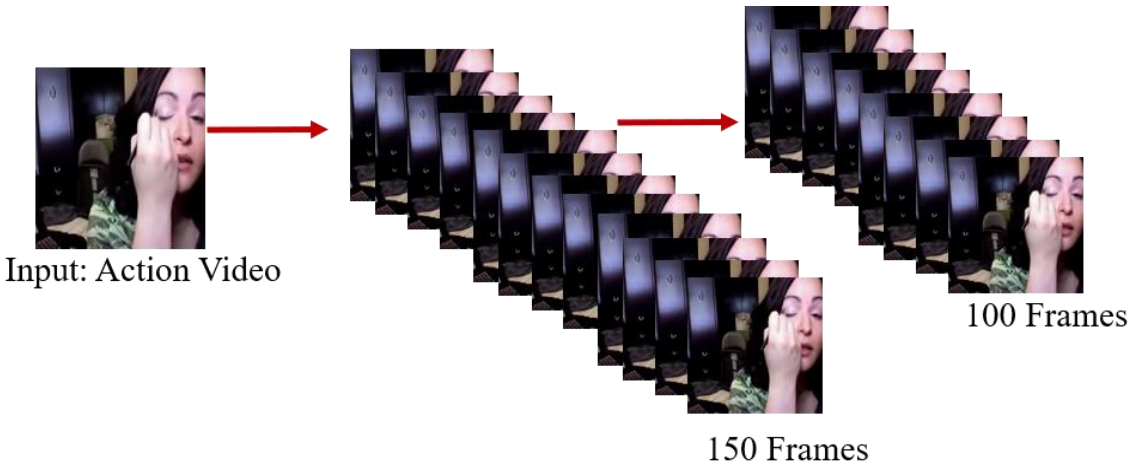


Figure 4.4: Sample of Key Frame Selection

Figure 4.3 and Figure 4.4 show the process of keyframe selection from video and example of the result obtained by the procedure respectively. The example shows that a sample of ApplyEyeMakeup action video with its frame sequences and the selected keyframe result.

4.3 Saliency Detection

Taking the extracted keyframes as an input, the saliency detection algorithm is employed to detect the salience region of frames.

To process the saliency detection, Pyramidal Guided Attention Network (PGANet)'s pre-trained model on object detection was used. This method of saliency detection was based on an attention-based pyramid module, which allows the network to focus on the region which are salient while manipulating information of multi-scale saliency and detection of edges saliency, which emphasis on salient edge information importance. Subsequently, it offers a strong cue for better segment salient objects and refinement of edge boundaries.

In the saliency detection, an additional role part of pyramid attention is that it offers mechanisms to efficiently improve the ability to represent corresponding layers of the network with receptive fields of enlarged. In the case of detection of salient edges, it paves the way for learning clear salient boundary estimation and so encourages better preservation of edge-based salient object segmentation.

4.3.1 Problem Formulation

As shown from most saliency detection developed in [52] and [53], for deep saliency-based object detection, the problem of salient feature preservation property in the salient edge is still unsolved. This means the current network architectures for saliency escalates to accumulate features of multi-layers. Although the final prediction of layers gets multi-level information and multi-scales as well as developing more clear segmentation of saliency, sharpening the edges or boundary between objects and the salient region is unsolved. This resulted from the nature of developed convolutional kernels and downsampling the result of spatial pooling.

Therefore, the concept of saliency detection is how to develop a clear refined object's boundary while detecting object saliency. This can be expressed as for input image I_k, Ground truth G_k and Salient object boundary map B_k):

$$Sal(I_K, G_K, B_K) = \frac{1}{K}\sum_{k=1}^{K}(I_K, G_K, B_K) \qquad 4.1$$

To solve the defined problem stated in section 4.3.1, it is better to have salient edge detection functions that learn clear salient edges by using the application of minimizing L2 norm loss function. It is expressed as:

$$\mathrm{Sal}\left(\mathrm{I_K}, \mathrm{G_K}, \mathrm{B_K}\right) = \frac{1}{\mathrm{K}} \sum_{\mathrm{k=1}}^{\mathrm{K}} \left(L^{Edge}(\mathrm{B_K}, \mathrm{E}(Y_{IK})),\ L^{Edge}\left(\mathrm{B_K}, \mathrm{E}(Y_{IK})\right)\right) \quad 4.2$$

$$\mathrm{Sal}\left(\mathrm{I_K}, \mathrm{G_K}, \mathrm{B_K}\right) = \|\mathrm{B_K} - \mathrm{E}(Y_{IK})\|_2^2$$

Where Sal, $L^{\mathbf{Edge,}}\mathrm{E}\ \mathbf{(Y_{Ik})}$, and *K* describes saliency, loss at edge detection, salient edge information, and the number of training sample**s** respectively**.** Using this loss function, the saliency result was obtained with a clear object boundary.

4.3.2 Saliency Detection using Pyramidal Attention and Edge Detection

The reason for selecting this SD method is that the saliency model representation of features is important. Additionally, it is required for efficient strategies in forthcoming features in a scale-spaces learning problem. This means, having the concept of multi-scale object saliency detection may produce better results since different objects may have different scale representations. Based on this idea, the pyramidal based approach overcomes such challenges resulted in many deep learning-based saliencies models.

Incorporation of attention mechanisms into the network can improve activities of selecting only relevant features. So, the model can only focus on interest regions and the mechanism of attention extending with hierarchical structure helps in saliency computation.

In the existing saliency detection methods, even with densely connected or top-down /bottom-up architectures in [53], [54],[38] were studied in salient object detection in a top-down fashion with a gradual approach, the problem of sharpness is still unsolved.

Therefore, an efficient method of improving sharpness is required. However, CNN is designed for producing hierarchical feature maps through repeated pooling, upsampling processes in which higher-layer gain larger receptive fields and able to stronger representation. But this may cause losing many details in spatial information which results in degradation of accuracy for lower-level tasks.

In the high-level tasks, pooling and upsampling features are good but not for low-level tasks such as salient object segmentation where clear pixel-wise activations are required especially for the boundary of salient objects. Therefore, saliency edge detection overcomes the challenges of locating salient edges and sharpen it.

4.3.3 Saliency Detection Algorithm

This algorithm shows stepwise saliency detection tasks for efficient detection of objects in the manner of focusing only on the richest information among the parts of scenes. It is described in Table 4.1.

Table 4.1: Saliency Detection Algorithm

Algorithm 2: Saliency Detection
Input**: Imge* ***Output**: Salient Image* ***im_res *// resizing images* ***batch_size*** *// set number of input data fed at one cycle*
Bigin

```
read image(data)
for i in range len(data): // read data
   img_name = read.name(os.path(data)
  original_img = read(data)
  copy = zeros(original_img.shape[0], original_img[1])
  res_img = preprocess (original_img, im_res)
  return [res_img], original_img, img_name
  x = Input(batch_shape=(1, im_res, im_res, 3))
  m = Model(inputs=x, outputs=sam_vgg(x))
  prediction =m. predict (res_img, batch_size=12)
 saliency = predict (res_img,
sam_vgg(original_img.shape[0], original_img[1])
cnvrt_to_rgb=saliency*original_image // converting to RGB
 return cnvrt_to_rgb
```

End

The algorithm of saliency detection uses the selected keyframe results as input and applies pyramidal attention guided for detecting salient regions of frames.

To process the task, convolution of features using VGG-16 and defines saliency for each convolved feature. Then attention map and edge detection were applied for better saliency

improvements. Finally, the concatenated edge and attention map developed for saliency convolution was given to SoftMax for predicting the final saliency result. The workflow of the saliency detection algorithm was described in Figure 4.5.

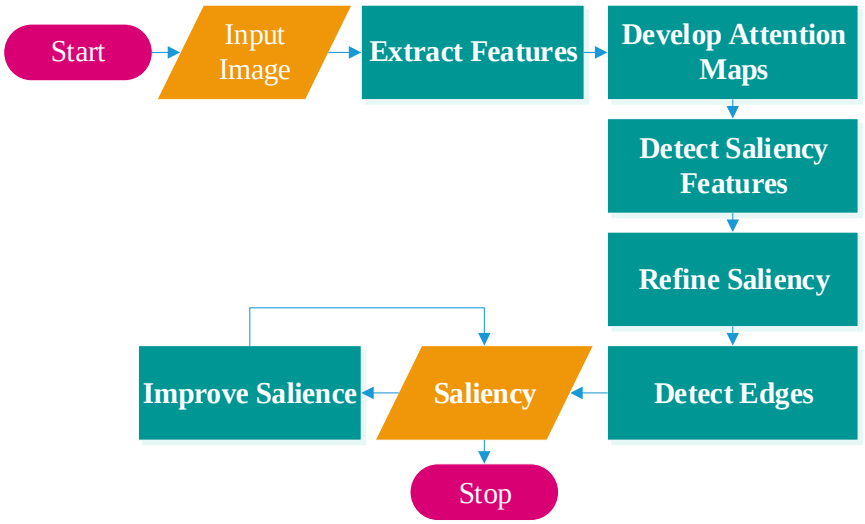


Figure 4.5: Workflow Process in Saliency Detection

From Figure 4.5, tasks in the saliency detection were conducted stepwise with consideration of each step's results. In the first step, input images are prepared for the next step to feature selection using VGG-16 as a backbone feature extractor. Then, for each selected feature, an attention map is developed by using multi-scale pyramidal attention to locate the exact salient features in the network. This is due to all features are not equally useful for representing salient features.

After clear attention maps developed by downsampling selected features and fed to sigmoid function, feature refinement is used to evaluate the initial saliency convolutional outputs. Then based on the refined features, the saliency object is easy to be detected using feeding the refined feature to directly sigmoid function.

The important and essential step in this study is that of clear boundary preparation for ease of understanding the salient object and the boundary, which is an edge detection step. Then saliency is now produced as a result of previous steps and one more needed task is improving the obtained result by using a concatenation of pyramidal attention results with cues of salient edges.

The final result of saliency detection is detected salient objects from up-sampled results of attention and edges for back to their original sizes. The obtained result was represented in Figure 4.6.

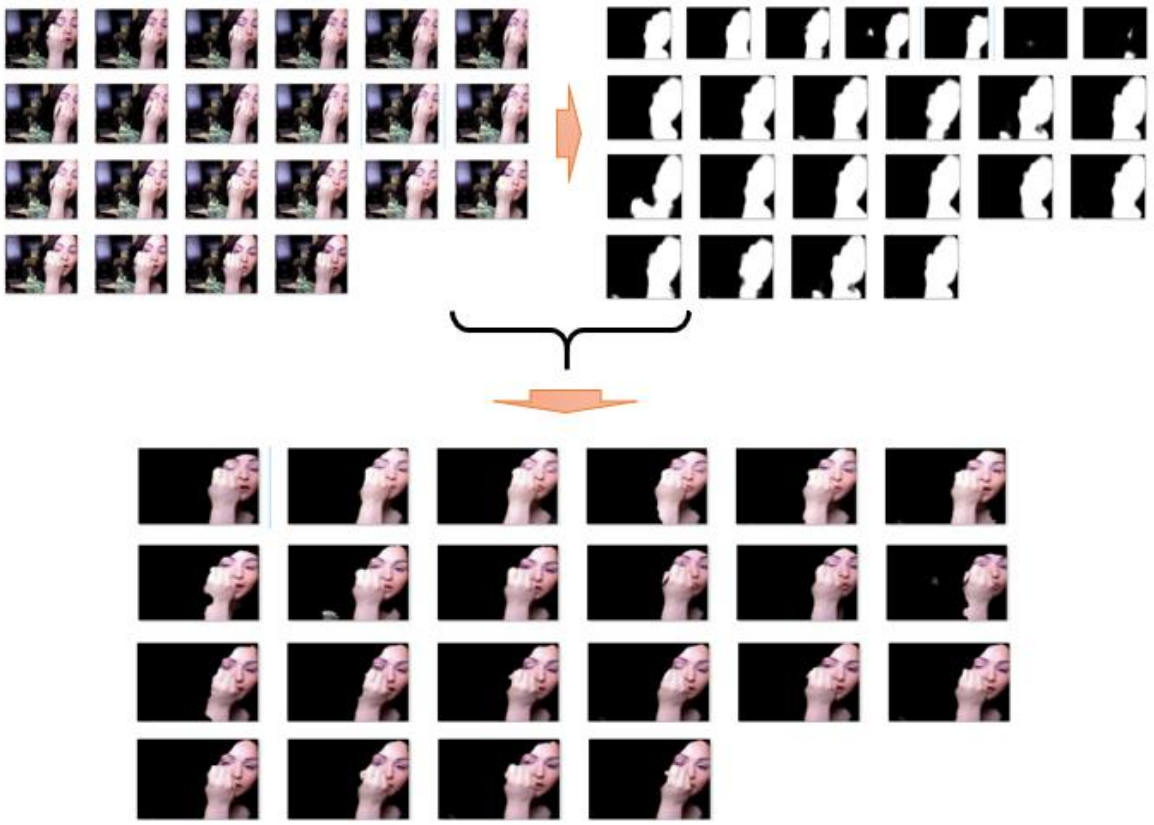


Figure 4.6: Sample of Saliency Detection for ApplyEyeMakeup Action

As shown in Figure 4.6, the first saliency results were binary information of frames which are black and white. By converting the binary information to color, the final clear information was obtained.

4.4 Overall Algorithms of KFSD for HAR

The summaries of algorithms used in this proposed study, key frame-based saliency detection for human action recognition using deep learning are those described in section 4.1 to 4.3. The algorithm starts with the approach of keyframe selection from sequences of the video in human action and then determines the potential frame from among others in the same action categories and selects it as a keyframe.

The process of selecting keyframe from video sequences and its challenges were described in section 4.2.2. Then even selected keyframe may contain irrelevant information such as

background clutter, motions, and other interferences that may result in difficulty of processing expected results and determining the type of action.

Therefore, the selected keyframe is again detected by the salient detection method and then fed to the action recognition network. The saliency detection algorithm was represented in section 4.3.3

The general concepts behind keyframe based saliency detection for human action recognition using deep learning proposed in this study are: First, video action is segmented into equal time dimensions (1FPS), and from the segmented frames, predicting keyframes is done by using absolute differences in their RGB of interframes.

Since the type of input video was color video and in the same way selected frames are also color frames then their local maximum computed. Second, the selected keyframes were selected for saliency detection which is used for focusing only relevant information while representing the type of action later.

The final network for representing the type of action is made up of bidirectional LSTM, which was used for determining sequence-based information. So, the reconstructed video of salient keyframes was consecutively fed to VGG-16 + Bi-directional LSTM, and based on learned sequences, the action was recognized.

4.4.1 KFSD-HAR Architecture Description

The architecture of proposed keyframe-based saliency detection for human action recognition consists of three main modules, namely: keyframe selection, saliency detection, and action recognition modules. This architecture is found in Figure 4.7.

A. Key Frame Selection Module

The keyframe selection module is used for selecting keyframe from video action sequences by calculating their inter-frame difference and compare with the local maximum. This module is expressed in section 4.2 of the proposed algorithm in detail. Figure 4.8 show the expected result of keyframe selection. In the module, video data were taken as input and converted to the sequences of frames. Next, frame difference was computed from the extracted frames. The the absolute difference of frames were compared to calculated local maximum of frames. Finally, frames those fulfill crateria to be keyframe were stored as keyframe, otherwise discareded. The following Figure 4.8 show all the concepts.

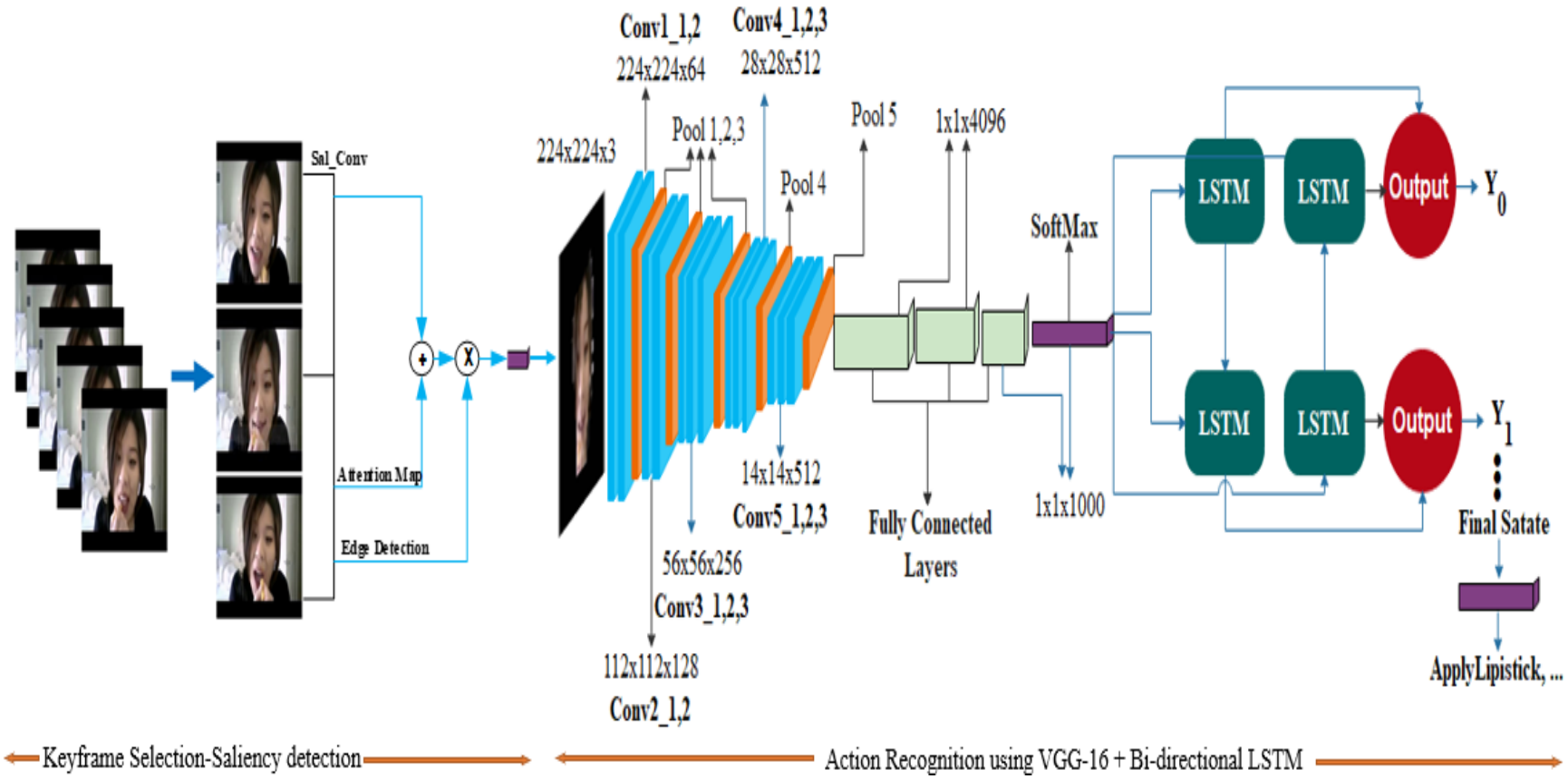


Figure 4.7: Architecture of Keyframe-Based Saliency Detection for HAR

Figure 4.8: Keyframe Selection Module Result

B. Saliency Detection Module

The saliency detection module detects salient information of selected keyframes done by the keyframe selection algorithm. This type of module broadly contains three main sub-modules(components) namely: attention module, an edge detection module, and a backbone network.

Attention Module: This module is based on the concepts of pyramidal attention which can be fused to produce representations of more discriminative features. This employs as the network focuses only on the most important features in regions rather than all parts.

The task of focusing on the important region and multi-task can be attained by attention architecture which is stacked like layers of multi attention developed on features of multi-scale forming combined pyramidal attention model. Similar to [55], Features of multi-scale can be obtained by downsampling features (X) from Conv-layers of the saliency network to multi-resolution gradually.

Let X be Conv-layer features consists of channel C with width W and height H, and then features are made up of ($X: \in \mathbb{R}^{W*H*C}$). So, we focus on learning attention masks which are equal in spatial size that outputs salient features X, based on information from multi-scale. Therefore, downsample X into multi-resolution with N steps, $X^n \in \mathbb{R}^{\frac{W^n}{2} * \frac{H^n}{2} * C}$ where n = 1, 2…N with N steps.

For X^n with n scale, the attention mechanism which predicts vital map, $\ell \in [0, 1]$ $\mathbb{R}^{\frac{W^n}{2} * \frac{H^n}{2} * C}$.

Regions in the input feature using SoftMax operation are used for determining the probability by which the model trusts the matching region.

It can be expressed as:

$$L^n = P(L = iX^n) \qquad 4.3$$
$$= \frac{\exp\,(W_i^n X_i^n)}{\sum_{j=1}^{\frac{W}{2^n}*\frac{H}{2^n}} \exp\,(W_j^n X_j^n)}$$

where i ∈ **1,…,** $\frac{W}{2^n} * \frac{H}{2^n}$ W_i^n are hidden layer weights which map to i[th] element of SoftMax location and L is a random variable which takes ℓ of $\frac{W}{2^n} * \frac{H}{2^n}$ values, ℓ is attention map, $\sum_i^{\frac{W}{2^n}*\frac{H}{2^n}} li$=1

Salient Edge Module: The obtained salient features need to be refined and the refined feature can be directly given to convolution layers stack with a sigmoid activation function. But detection only cannot give a strong boundary between background and salient object. This initiates to develop an edge detection module that facilitates the network to focus on the alignment of salient boundaries and understand to refine saliency maps by using the information of salient edges.

Let us develop the salient edge detection module as $\mathcal{E}(Y_{ik})$ by using training data ($B_{k,}$ $I_{k,}$ $G_{k,}$), where, Y is refined saliency feature, I_k color image input, B_k boundary of salient edge, and G_k saliency map ground truth, then it can be learned using reducing loss function of L2 norm.

$$\mathrm{Sal}\left(\mathrm{I_{K,}G_{K,}B_K}\right) = \frac{1}{\mathrm{K}}\sum_{\mathrm{k=1}}^{\mathrm{K}}\left(L^{Edge}(\mathrm{B_{K,}E}(Y_{IK})),\ L^{Edge}\left(\mathrm{B_{K,}E}(Y_{IK})\right)\right)$$

$\mathrm{Sal}\left(\mathrm{I_{K,}G_{K,}B_K}\right) = \|\mathrm{B_K} - \mathrm{E}(Y_{IK})\|_2^2$, where Sal is saliency and L is the loss at edge $\boldsymbol{E(Y_{ik})}$. 4.4

Backbone Network: This network is developed from the VGG-16 model. It is preferred due to its usage in many models of saliency and simplicity. From the layers, only the first five convolution layers are used. For preserving more spatial information, the last pooling layer, pool5, is deleted. This network is used for extracting features from frames which facilitates ease of later tasks done.

By using the architecture with its all algorithms, the key frame-based saliency detection result was achieved. However, the obtained result looks like simple salient object detection which does not give us essential information.

Note that, the goal is recognizing action from selected and saliently detected keyframes from video sequences. But still, the obtained result does not provide us what type of action information that the selected frame contains. This is due to the nature of saliency detection algorithms provides binary information (0 or 1), which is black and white color.

Additionally, since the selected keyframe from videos has irregular shapes or patters, it is not preferable for even representing the exact shape of the object. This is one difficult faced while implementing the network. Moreover, the absence of ground truth saliency for selected frames represents in accurate detection of salient information for all frames. This difficulty is that most developed saliency models are for a static object which has mostly regular patterns or shapes with ground truth.

To overcome such challenges, one more contribution to this study is that producing a new object detection algorithm by using the information of originally selected keyframes and their saliency result obtained. This was done by multiplying the original frame with its corresponding saliency. However, the result obtained by this method again does not give us relevant information. So, one more step is to thresholding the saliency result before multiplying it with the original keyframe. Then the better result is obtained in color for most regular patterned object action. Figure 4.9, shows the steps and results gained by each method.

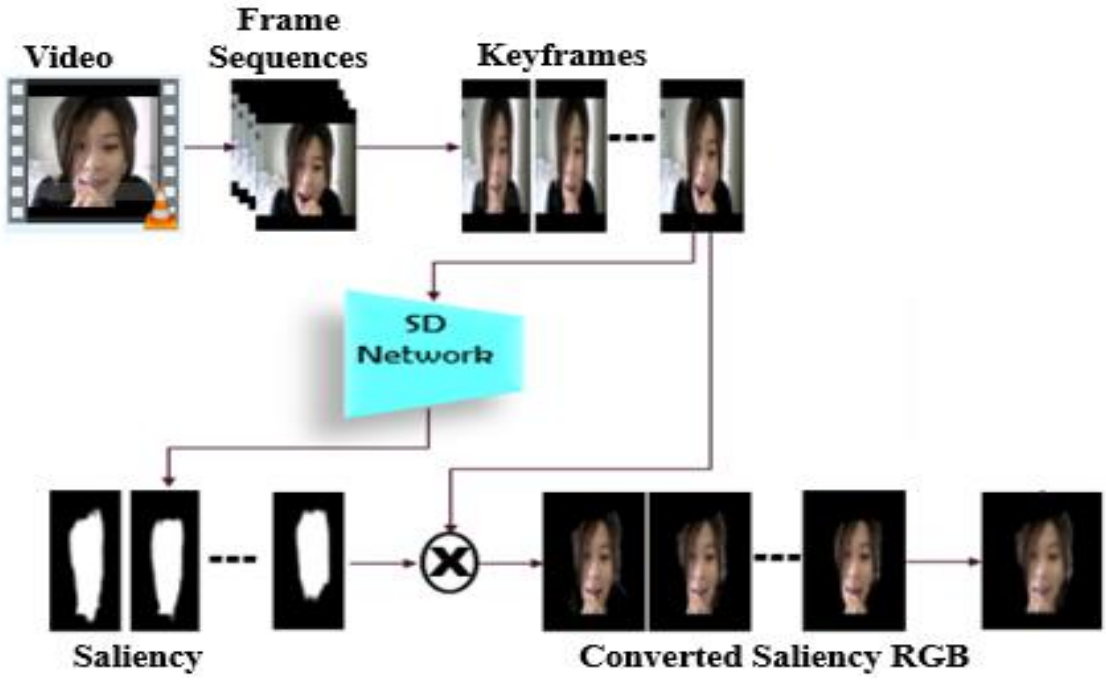


Figure 4.9: Saliency Detection Module Result

As represented in Figure 4.9, the process of saliency detection in this study is based on the keyframe selection method and it takes the selected keyframes as input to the saliency detection network. Then, after applying saliency detection on keyframes, the output was obtained and improved using the conversion of the output to RGB color information.

C. Action Recognition Module

This is the third type of module of KFSD HAR architecture described in section 4.4.1. It is the second network, next to SDN (Saliency Detection Network), which is used for recognizing the type of action performed. It contains two networks namely: CNN and LSTM(Bi-directional).

CNN: CNN (Convolutional Neural Network) is used for storing frames into channels of inputs of CNN and then classify the image or stacked video frames in the channels. However, training deep learning models for such action representation from video requires huge computation and high processing power for weight adjustment of the CNN model.

Getting the required model using such a strategy is an expensive process. But, to overcome such challenges, the approach of transfer learning in which pre-trained parameters of VGG-16 on the ImageNet dataset were used in this proposed for extracting frame's features.

VGG-16 + Bi-directional LSTM: VGG-16 is used for extracting features from each video frame. Then the sequences of selected frame features are fed to bi-directional LSTM for sequence learning. Then based on the learned sequences, classification of the type of action was made.

The preference of bidirectional LSTM rather than LSTM is that it enhances model performance on sequence classification challenges. It trains two LSTM on the input sequence where time steps of input sequences are available. The initial input sequence a - is and the second is a reversed copy of the input sequence.

Generally, with bidirectional LSTM, it is possible to run inputs in two ways, one from past to future and the other from future to past instead of only passed information of LSTM. Therefore, it could be easy for preserving information from the future and past. Additionally, it is pretty for understanding the context in a good manner.

Figure 4.10 shows the result obtained by the action recognition model at the final. Figure 4.10, shows that the action recognition module takes the output of the saliency detection result that is reconverted salient keyframes to video. Next, action recognition extracts feature

from videos using VGG-16 and learn the sequences for classification of actions using Bi-directional LSTM. Then the learned sequences were used for predicting the state of the action which is used for predicting the type of action. Finally, the action was recognized from the final state of action.

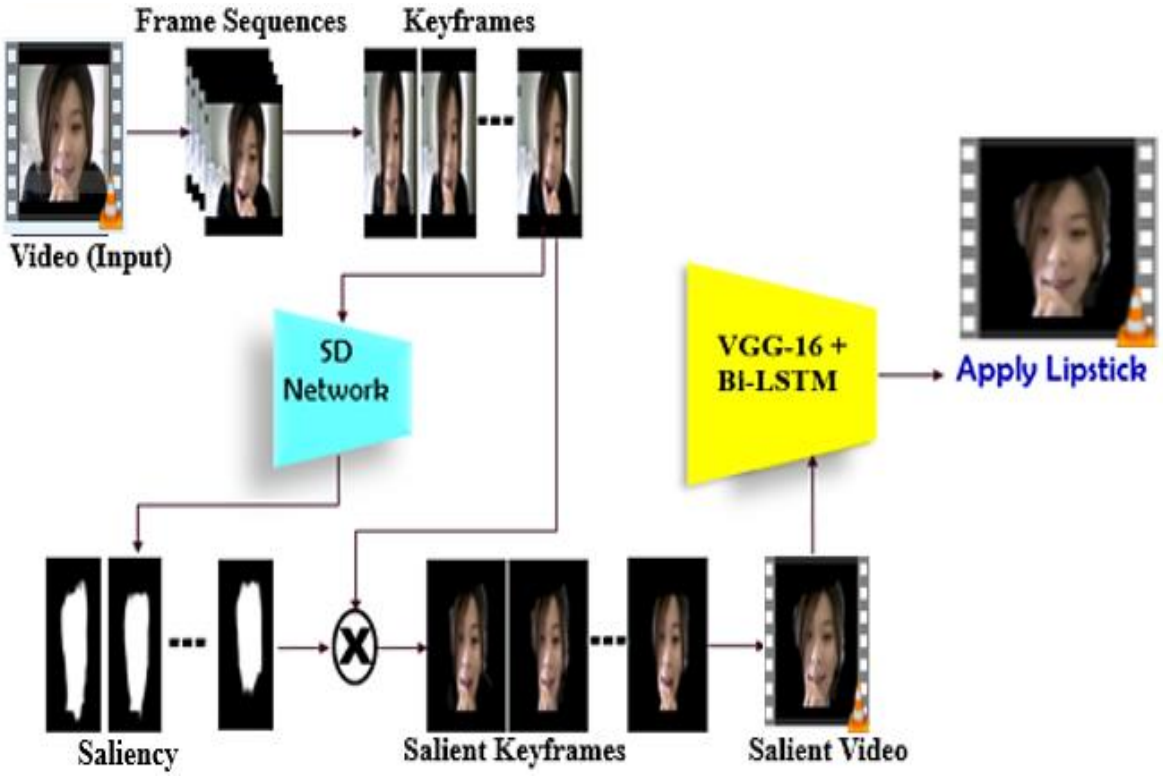


Figure 4.10: Action Recognition Module Result

The selected model trains the network on the prepared data by using developed classifier network defined. This classifier has a pretrained weight model which is loaded for training. Then the loaded weight of pretrained model was fit to the prepared data with required parameters.

Finally, a predifened classifer is used to predict the prepared data. The predictor loads a pretrained model and scans the video files in the path. If the list of data are available, it shuffles them and scans the actual class. Then it predicts the action based on pretrained classifier and calculates accuracy of prediction.

4.5 Summary

In this representation of the proposed algorithm, the concept behind the need for a specific algorithm for the specified problem was described. An algorithm such as keyframe selection for selecting relevant frames from sequences that have potential information to represent the action, salience detection algorithm for detecting salient information from where the action mostly attracts attention, and sequencing all the processed data to recognize the type of action using action recognition algorithm is discussed.

Generally, keyframe selection-saliency detection is a sequential process done by a separate algorithm using interframe difference based on RGB compared with computed Local Maximum information, then detecting salient features was performed by salience detection network followed by recognizing the type of action based on processed information using action recognition model (VGG-16 + Bi-directional LSTM)

CHAPTER FIVE

5. IMPLEMENTATION OF PROPOSED KEYFRAME-BASED SALIENCY DETECTION APPROACH

5.1 Introduction

In this chapter, the implementation of the proposed keyframe-based saliency detection for human action recognition using deep learning was described with the obtained results. For implementing the proposed approach, the environment were set as described in chapter three. These setup includes: python programming language, keras framework backend as tensorflow, OpenCv library, Pandas, Scikitilearn, Seaborn and etc. The experiment was performed on Dell Optiplex 3040 with 8 RAM CPU, GeoForce RTX Super 2070 with Hydro G 650W Power supply.

5.2 Keyframe-based Saliency Detection Model Implementation

The step followed for implementing the proposed approach was given in Figure 5.1.

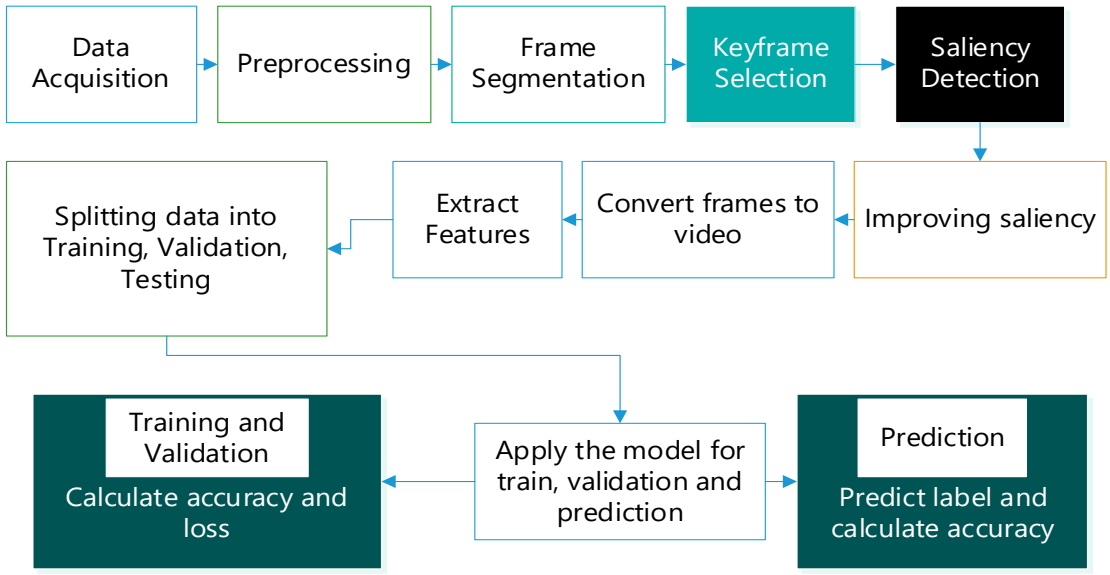


Figure 5.1: Keyframe-based Saliency Detection implementation steps

5.3 Descriptions of Proposed Approach and its Outcome

The developed approach for implementing keyframe-based saliency detection for humanction recognition includes:

***Data acquisition**:* selected data of action video were required to sart the process of implementation. These data contains 20 class of actions with total 2850 videos.

***Preprocessing**:* the obtained video were resized (320 x 240), width, height respectively based on the requirements of input model. Additionally, padding, rescaling were performed.

***Frame segmentation**:* input video were segmented in to equal frame of 1second and stored in order. Reading multi input and process was the contribution made in which existing reads only one input and process at atime.

```
path = "C:/../"           // input Video path
framepath = "C:/../”    // segmented frame frame path
folder = os.listdir(path)
for _ in range(len(folder)):
     os.mkdir(path + '/' + folder[_]),os.mkdir(framepath +
'/' + folder[_])
def converter(videopath,inpath,folder_name):
    images = []
    cap = cv2.VideoCapture(path)
    curr_frame = None
    prev_frame = None
    frame_diffs =[],frames = []frame_diffs =[],frames = []
    success, frame = cap.read()

    cv2.imwrite(framepath+ ’/’+ “frame_”%d, frame)
```

Keyframe selection: keyframes were selected from the segmented frames using absolute difference of RGB between consucutve frames and the computed local maximum. The problem sequence ordering in existing keyframe selection was solved as a major contribution to this process.

```
keyframe_id_set = set()
USE_LOCAL_MAXIMA:
        ##print("Using Local Maxima")
        diff_array = np.array(frame_diffs)
        sm_diff_array = smooth(diff_array, len_window)
        frame_indexes = np.asarray(argrelextrema(sm_diff_array,
np.greater))[0]
        for i in frame_indexes:
            keyframe_id_set.add(frames[i - 1].id)
cap = cv2.VideoCapture(str(framepath))
    curr_frame = None,  keyframes = [], success, frame =
cap.read()
    idx = 0
    while(success):
        if idx in keyframe_id_set:
```

```
            cv2.imwrite((Keyframe_Path + folder_name[:-4]+
"001_{0:06d}.jpg".format(int(idx))), frame) //Sequence ordering
```

The result of above steps were given in Figure 5.2.

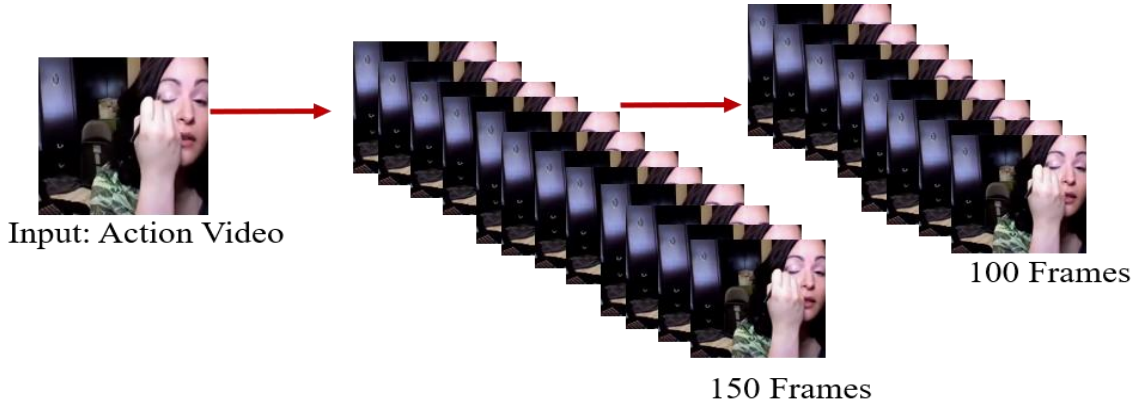


Figure 5.2:Keyframe selection from segmented frames

Saliency detection: this was performed by detecting the salient area of frames to represent the motion of action. Pyramidal guided attention network for saliency detection is used for this process. The obtained saliency result of action was improved using conversion of binary information to RGB color as contribution made to the step.

```
def saliency(images):
    seed = 7
    random.seed(seed)
    test_data = []
    for i in range(len(images)):  // data input
        Ximg, original_image = get_test(images[i])
        predictions = m.predict(Ximg, batch_size=12)//
        res_saliency = postprocess_predictions(predictions[9][0,
:, :, 0], original_image.shape[0], original_image.shape[1])
        x=(res_saliency)>150 //

backtorgb=cv2.cvtColor(x.astype('uint8'),cv2.COLOR_GRAY2RGB)
        multiply= (backtorgb*original_image)//
```

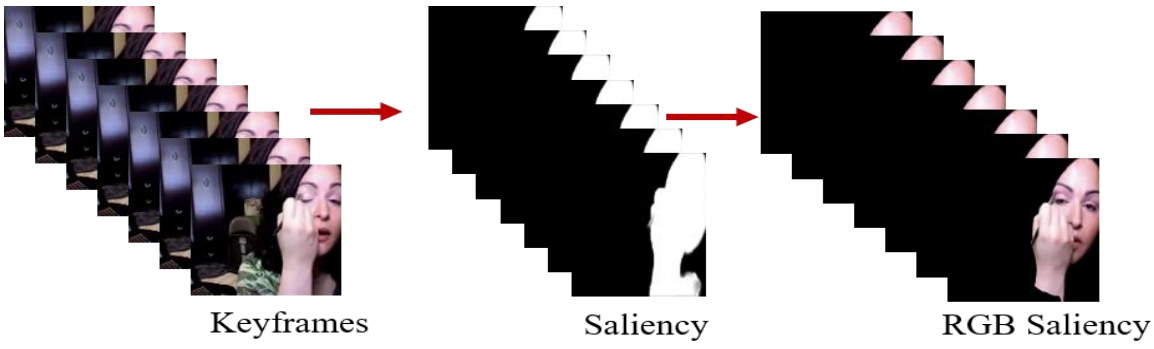


Figure 5.3: Saliency detection from keyframes

Action recognition: in this step, the developed model were trained using action recognition network, VGG-16+Bi-directional LSTM. It is based on the extracted features ofsalient keframes and the learned sequences of consecutive frames in backward and forward directions. The training sample code and results were given in chapter six. For sample of action prediction let use this sample code and the results as below:

```
from keras_video_classifier.library.recurrent_networks import
VGG16LSTMVideoClassifier
from keras_video_classifier.library.utility.ucf.UCF101_loader
import load_ucf, scan_ucf_with_labels

count = 0, correct_count = 0
for video_file_path in video_file_path_list:
        label = videos[video_file_path]
        predicted_label = predictor.predict(video_file_path)
        print('predicted: ' + predicted_label + ' actual: ' +
label)
        correct_count = correct_count + 1 if label ==
predicted_label else correct_count
        count += 1
        accuracy = correct_count / count
        print('accuracy: ', accuracy)
```

Figure 5.4: Action prediction (testing) result

This result of action prediction was implemented with selecting five classes of prepared data which contains 500 videos and applied VGG-16+Bi-LSTM prediction.

CHAPTER SIX

6. RESULT AND DISCUSSION

6.1 Introduction

This chapter describes the outcome of the proposed keyframe based saliency detection method for human action recognition. It also represents the result obtained from the implemented algorithms of the proposed system. The result obtained from the implemented algorithms at both training and testing of the proposed method was also presented using conducted experiments. Moreover, the discussion of the obtained result of the proposed network was presented. Generally, in this chapter results obtained by application of keyframe-based saliency detection for human action recognition were presented with discussion in a compact and informative manner.

6.2 Experimental Result

The experiment was conducted on anaconda python 3.7 with OpenCV 4.2 and TensorFlow 1.14 framework using Keras 1.13 backend. The experimental procedure was separated into two processes as described in chapter 3. The first procedure was keyframe selection followed by saliency detection and the second procedure was, action recognition, which is the stated action class from previous procedures. The detailed description of the procedural steps was given in section 5.2.1.

6.2.1 Keyframe Selection and Saliency Detection

For this stage, from the UC101 dataset, 20 action categories were selected, where each category contains video ranging from 100 to 200. From each category, the keyframe was selected using a keyframe selection algorithm. Then, once the keyframes were selected from the categories, saliency detection was applied using a pre-trained model of Pyramidal Attention Guided Network on Extended Complex Scene Saliency Detection dataset by VGG-16. The saliency detection stage takes the selected keyframes as an input, where some parts of the pre-trained PAGNET network was considered for the detecting salient regions of keyframes

During experimentation, it was noticed that the results of saliency detection show binary color information which may be difficult for the feature extraction process while sequence

learning stage in Bi-directional LSTM using VGG 16 network. This is because the input for VGG 16 network considers 3 color channels (RGB color).

To alleviate this challenge, this study contributed to changing the binary information of data to color data which was most usable while extracting the features. The processed result was given in Table 6.1.

Table 6.1: Keyframe Selection-Saliency Detection Result

No	***Action Category***	***Total Video***	***Sequences***	***Keyframes***	***Saliency***
1	ApplyLipistick	114	22,800	11,478	11,478
2	Bandmarching	155	31,000	15,481	15,481
3	BaseballPitch	150	30,000	13,292	13,292
4	BenchPress	160	32,000	14,101	14,101
5	Billards	150	30,000	15,346	15,346
6	BlowDryHair	131	26,200	13,031	13,031
7	Bowling	155	31,000	15,098	15,098
8	BoxingSpeedBag	134	26,800	13,632	13,632
9	HorseRiding	164	32,800	16,517	16,517
10	HaulaHoop	125	25,000	11,802	11,802
11	IceDancing	158	31,600	16,108	16,108
12	JampingJack	123	24,600	10,162	10,162
13	JampRope	144	28,800	15,281	15,281
14	Kayaking	141	28,200	14,292	14,292
15	Knitting	123	24,600	12,536	12,536
16	PlayingDhol	164	32,800	16,924	16,924
17	PlayingGuitar	160	32,000	16,420	16,420
18	PlayingTabla	111	22,200	11,437	11,437
19	Typing	136	27,200	13,765	13,765
20	WritingonBoard	152	30,400	15,103	15,103
Total		**2850**	**570,000**	**281,806**	**281,806**

As indicated in Table 6.1, from the total number of 570,000 frame sequences of selected 20 categories UCF 101 dataset, 281,806 number keyframes were selected using keyframe

selection based on absolute frame difference compared to the computed local maximum method (as indicated in 5th column of Table 6.1). The same number of frames were considered for saliency detection (as indicated in the 6th column of Table 6.1). The input for saliency detection is the selected keyframe and the output of saliency detection was detected frame's action region, and they all are binary images(frames).

The binary color output is irrelevant for the proposed work and hence, binary color frames were purposely transformed into a color image. This color transformation is employed because the selected VGG 16 networks for feature extraction uses an RGB image. After the saliency detection and color conversion, the reconstructed color frames output became reconverted to video with the only salient region. As shown in Figure 6.1, the visual representation of the selected keyframe, the detected region of interest, and the colored conversion process are presented. The final output is a video frame with a region of interest and can be transformed into a video.

As represented in Table 6.1, the result of selected total keyframes from expected frame sequences was less than half. This can be obtained by subtracting total frame sequences from the total number of selected keyframes. It is 570,000 - 281,806 which gives 28, 8194. Finally, these selected keyframes were used for training the network to represent the action.

6.2.2 Keyframe-based Saliency Detection for Human Action Recognition

Action recognition in this step was performed by grouping salient keyframes selected from the original action video of 20 classes and then reconstructed video to represent action using VGG-16 + Bi-Directional LSTM. A sample of obtained results was given in Figure 6.2. To evaluate the obtained results by applying the developed KFSD HAR method, pre-determined evaluation criteria which are accuracy and loss computation at both training and validation was undertaken. A graphic representation of accuracy and loss for both training and validation was given in Figure 6.3.

From Figure 6.3, initially, the loss was high which represents the model does not understand data, but after the 5th epoch, it becomes decrease. On the other hand, accuracy was increasing as the number of epochs increases. The result of the newly proposed model based on obtained accuracy and loss during both training and validations using UCF-101 of 20 classes at 28th iteration was given in Table 6.2.

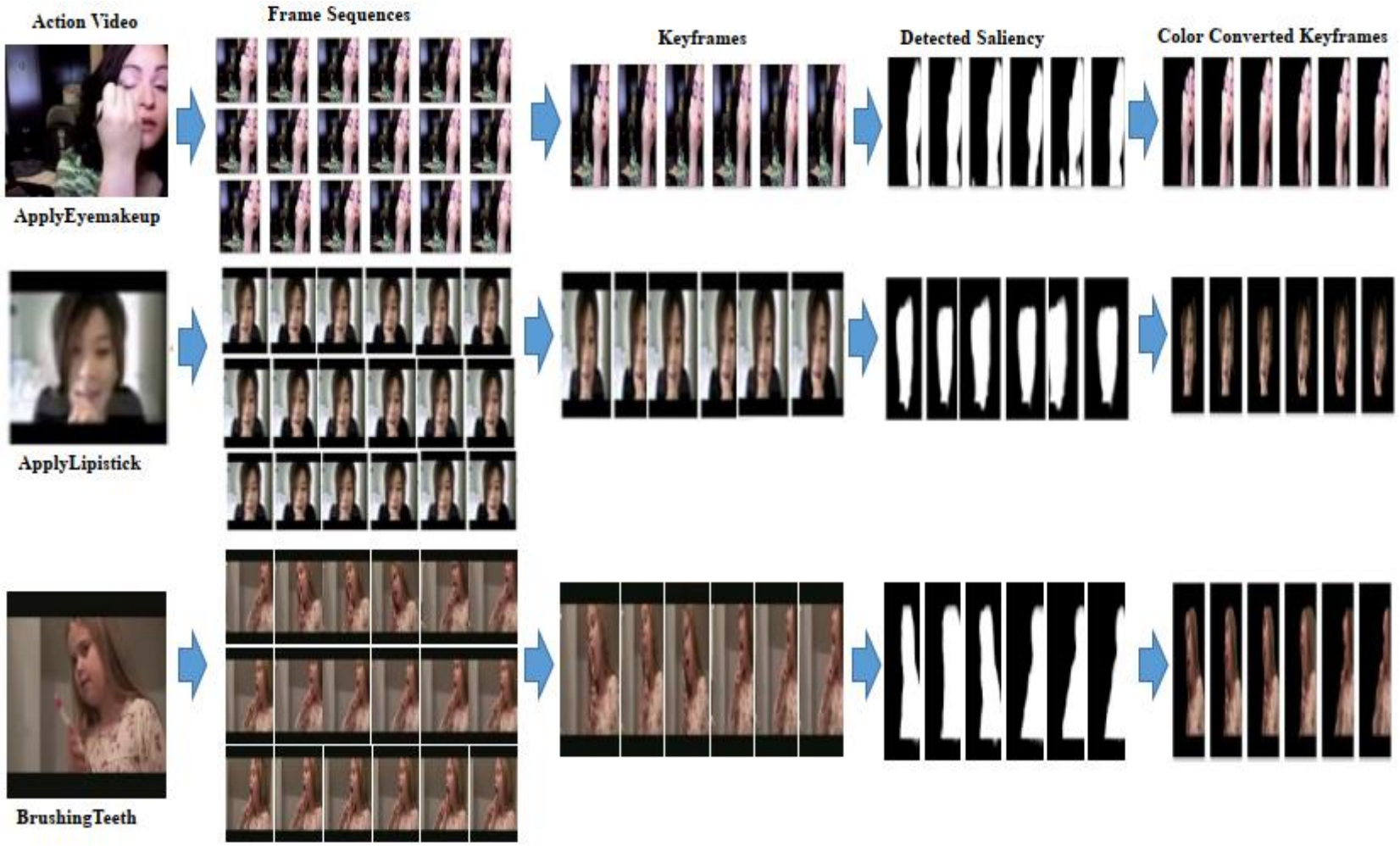


Figure 6.1: Keyframe Selection-Saliency Detection Sample

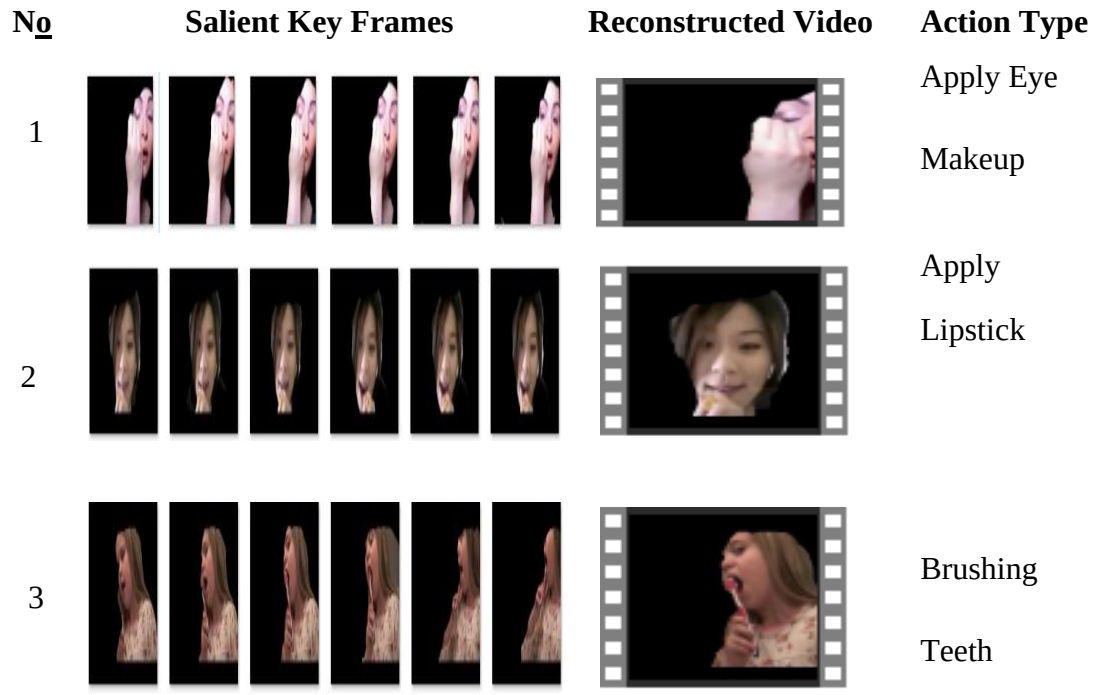


Figure 6.2: Sample of Action Recognition using KFSD HAR

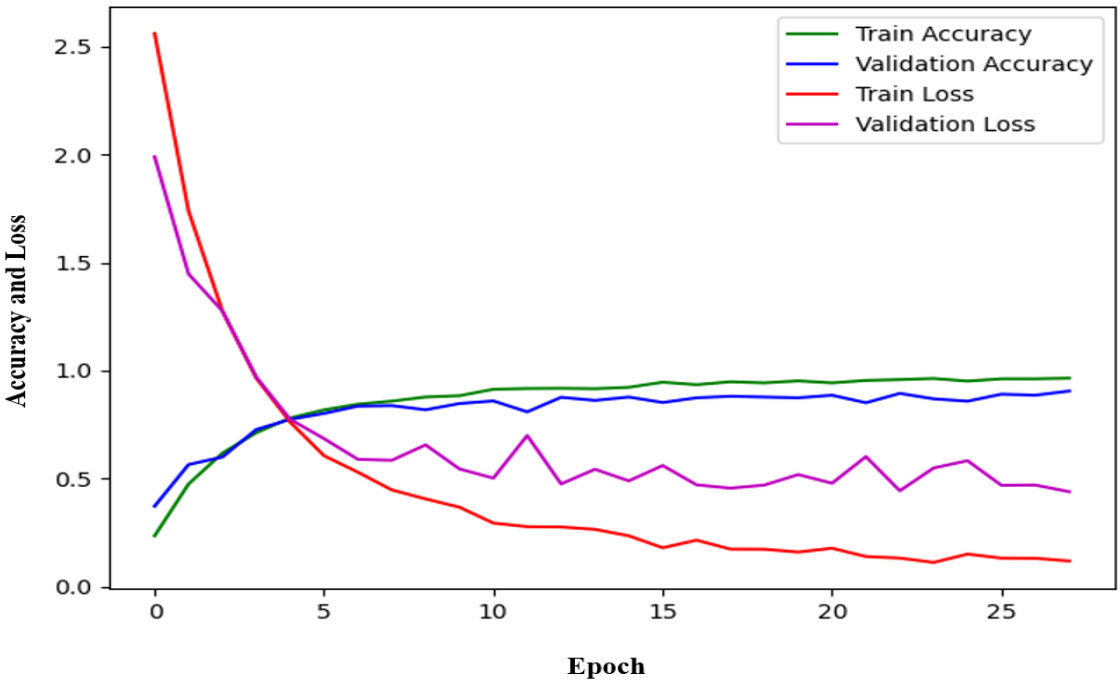


Figure 6.3: Graph Representation of Accuracy and Loss of KF-SD HAR

As Figure 6.3 shows, proposed KFSD HAR accuracy and loss for training and validation. As the result represents, the model starts to perform better accuracy and validation after 4th

epoch and continues to improve the result until the final 28th epoch. This describes the proposed model is good improving the accuracy and loss for both training and validation.

Table 6.2: Proposed KFSD HAR Mode Result

Model Name	***Accuracy and Loss at training and validation***			
	Training		***Validation***	
	Accuracy	***Loss***	***Accuracy***	***Loss***
KFSD HAR	0.9647	0.1188	0.9050	0.4394

Table 6.2 shows that the computed loss of the training data was much lower than the accuracy obtained. This shows the proposed model in fine for training the data in a good manner. For accessing the performance of the developed model result, a confusion matrix was developed. For representation, the predicted and actual class descriptions were given using Figure 6.4.

6.3 Comparison of KFSD HAR and VGG-16 + Bi-LSTM Model

This section represents the distinction between the existing and newly proposed model results. It was based on the performance evaluation used in the training and validation stage done in this work. First, selecting 20 classes of action videos from UCF-101, and a total of 2850 video actions were selected and applied keyframe selection. Then a total of 281,806 keyframes were used for saliency detection and salient keyframes were converted to RGB and forms video. Then the result was evaluated with the existing method. Finally, the result shows that there is an improvement in the proposed method based on evaluation metrics. The result was given in Table 6.3, Table 6.4, and Figure 6.5 respectively.

Table 6.3: Result of Existing and Newly Proposed KFSD HAR Model

Model Name (Existing vs Proposed)	***Accuracy and Loss at training and validation***			
	Training		***Validation***	
	Accuracy	***Loss***	***Accuracy***	***Loss***
CNN	5.65	152.08	4.57	153.82
VGG 16 + LSTM	66.73	100.26	71.88	91.39
VGG-16 + Bi-Dir. LSTM	96.37	12.31	88.70	48.96
KFSD HAR (Proposed)	**96.47**	**11.88**	**90.50**	**43.94**

As shown in Table 6.3, the proposed model, KFSD HAR outperforms the stated three models using training results at the 28th epoch. This was due to the developed model used proposed concepts of keyframe using and detecting salient regions of those frames which facilitates processing time and computational requirements.

Table 6.4: Comparison of Proposed Model and Existing Using Training Time

S.No	***Model Name***	***Overall training time/sec. and accuracy***	
1	CNN	329	4.023
2	VGG 16 + LSTM	175	53.193
3	VGG-16 + Bi-Directional LSTM	66	82.724
4	**KFSD HAR (Proposed)**	**61**	**85.964**

From Table 6.4, the overall training time for completing the training process by CNN, VGG-16 + Bi-Directional LSTM, and proposed KFSD HAR models were given. As shown in Table 6.4, compared to the three stated models, KFSD HAR takes less time to complete the training. This represents, it outperforms the stated models since the selected keyframes from videos and applied saliency detection were performed, and the process of training is fast.

Table 6.5: Comparison Between Existing Keyframe Selection and Proposed Method

S_No	***KF Methods***	***Expected KF***	***Accepted KF***	***Drop rate (%)***
1	Cross Correlation	144	10	93.06
2	Optical Flow	144	4	97.22
3	Sharpness	144	133	7.64
4	**KS (Proposed)**	**150**	**101**	**6.67**

From Table 6.5, the difference obtained by the experiment conducted between existing and newly proposed keyframe selection method using one action class was described. Even if the proposed method accepts more frames, it outperforms while checking the result.

Moreover, even if existing keyframe selection methods defined in Table 6.5 were showing a lower accepted keyframe, they didn't produce a better result. This can be shown poor video sequence while playing, very fast (example Optical Flow-based), very slow (example, sharpness based) were reduced the quality of video information and sequence consistencies. In case very fast motion which is produced from very low frames may reduce video

information and very low motion can cause sequence inconsistency which shows poorly organized frames.

However, in the newly proposed method, the produced video has clear information and looks better compared to the sharpness, cross-correlation, and sharpness keyframe selection methods.

Generally, the newly proposed keyframe selection based on absolute color differences of frames using computed local maximum performs better results while compared to other methods defined in Table 6.5.

The evaluation was done based on experimental results with defined parameters and additionally, separate testing results after reconstructing video from selected keyframes. Such evaluation was video quality, frame sequence consistency, and clarity of the constructed video.

Moreover, for assessing the performance proposed keyframe-based saliency detection model, a confusion matrix is used. It shows the number of the true label of classes which is named as an actual class and the predicted label. It is obtained from the model prediction result implemented using a developed prediction algorithm in KFSD HAR. For implementing the model prediction, 5 class of data where each class contains 100 videos and forming 600 video data were used.

As shown in Figure 6.4, the model predicts Billards, HorseRiding, and PlayingDhol's actions perfectly. The two actions, ApplyLipistick and JampingJack were classified as JampingJack and Horse Riding respectively with less value. This shows the proposed model is good for predicting correctly detected salient frames. On the other side, saliency detection has some limitations for irregularly patterned frames, complex and low visualized scenes. This was due to the original UCF-101 dataset was very challenging and has such difficulties.

The action categories were collected from a real environment and challenges such as similar background, camera motion, and action similarity. Such challenges, especially, visual similarity between the object and its background made the saliency detection model detect the region with low consideration. Moreover, the visualization of objects per scene and shape affects saliency results.

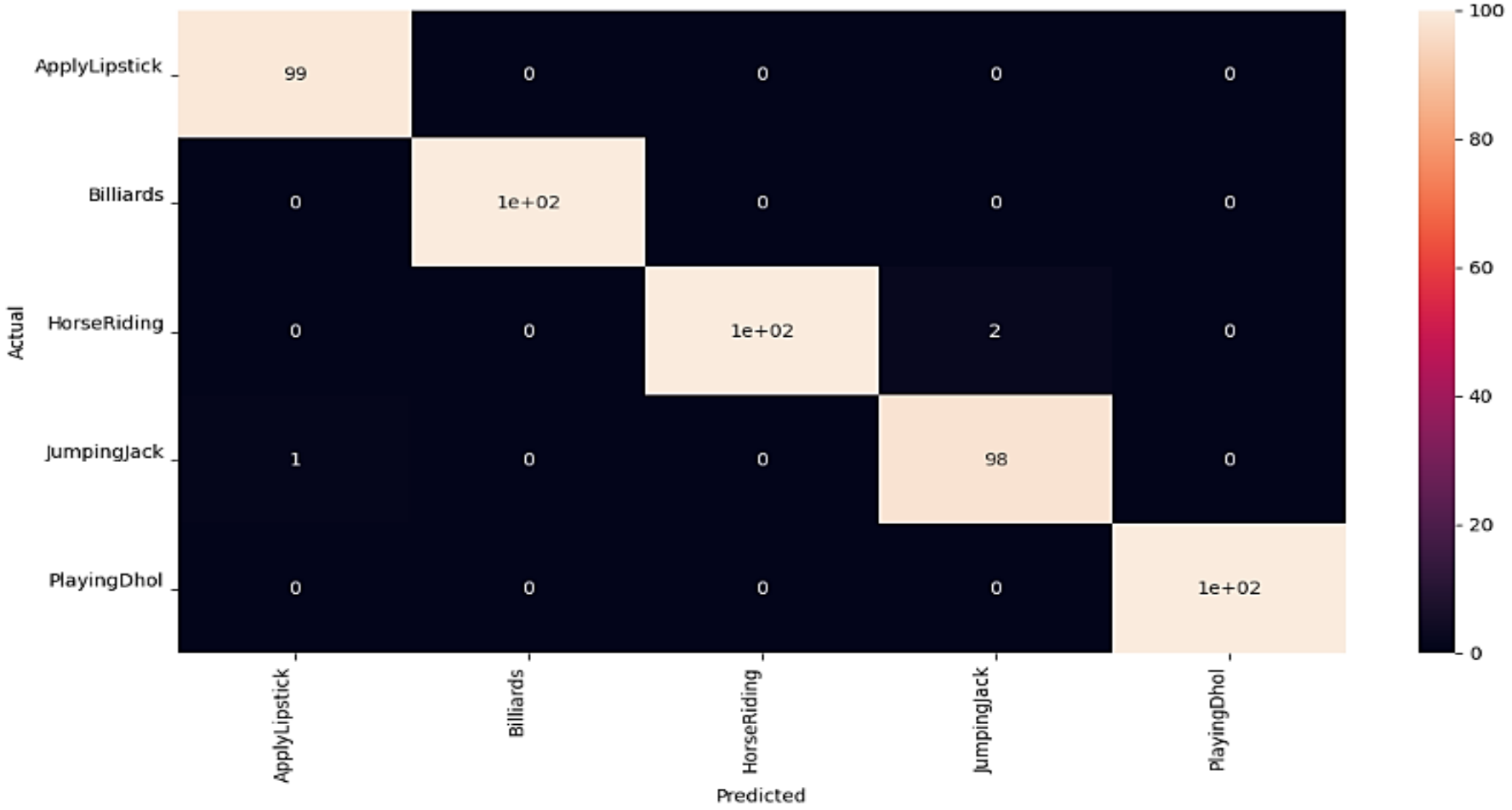


Figure 6.4: Confusion Matrix for Proposed KFSD HAR Result

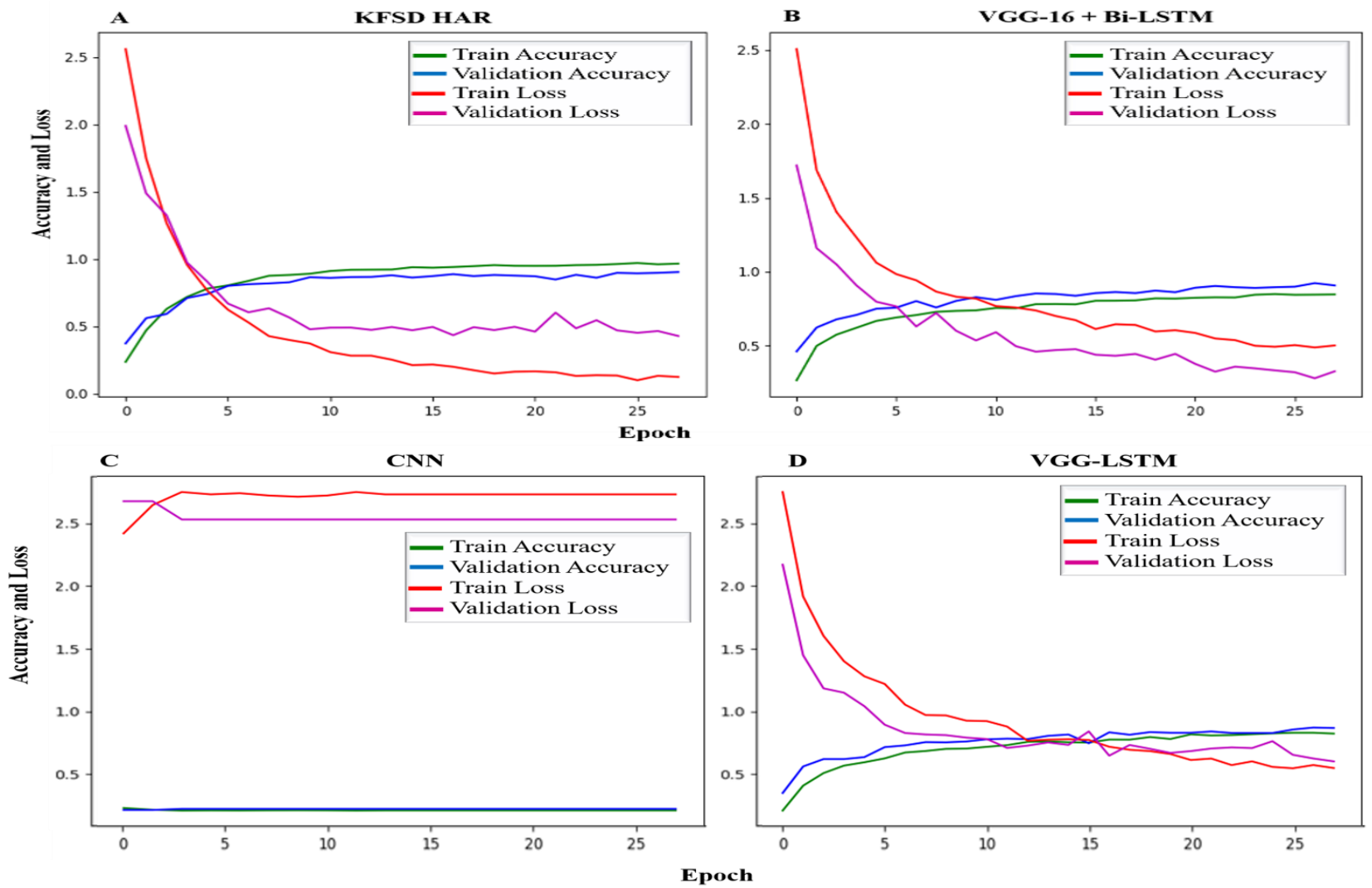


Figure 6.5: Comparison of KFSD HAR, CNN, VGG-LSTM, and VGG-Bi-LSTM

From Figure 6.5, the accuracy of training and validation were increasing as the number of iterations increases in the case of the newly proposed model in A and loss for training and validation was decreasing. On the other hand, in the existing model in B, the accuracy of training was increasing, while the accuracy of validation started to decrease after 25 epochs. Additionally, training and validation loss was increasing especially after 24, 25 iterations respectively. In C**,** accuracy and loss were separated in which loss was very high, and accuracy was very low. Such a model, CNN, was very poor training the data and it was not recommended for predicting the action. Finally, in D validation accuracy is greater than accuracy and both losses were higher compared to the proposed new model, A. This represents the newly proposed KFSD HAR model outperforms the existing CNN, VGG-LSTM, and VGG-16 + Bi-directional LSTM models.

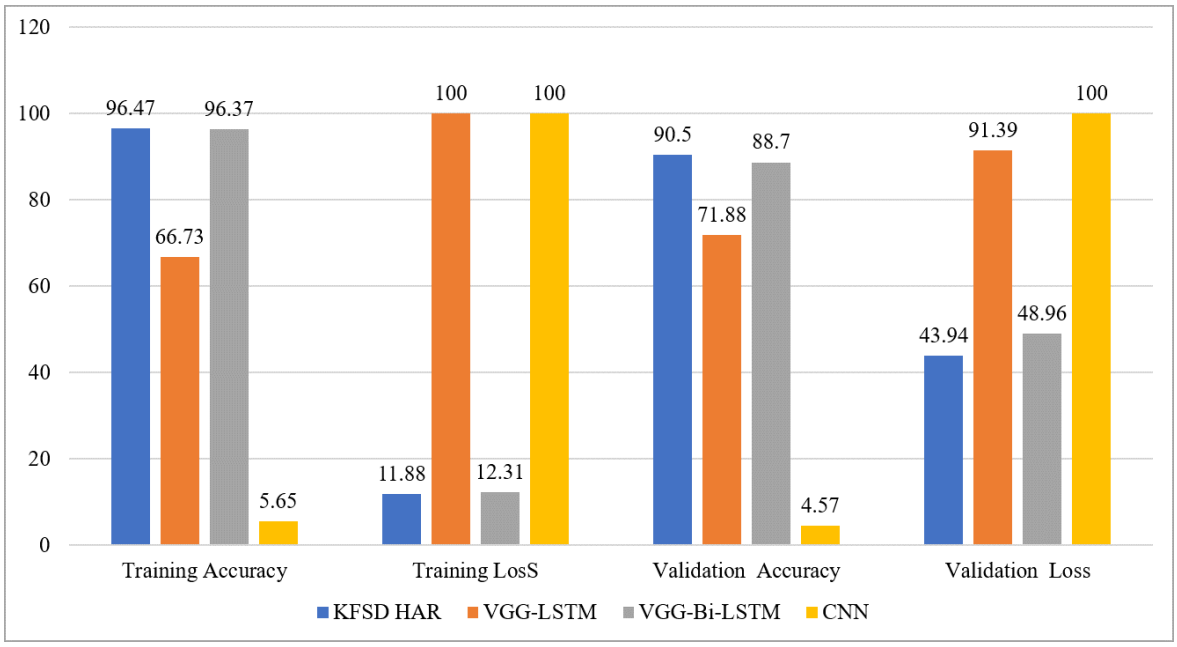


Figure 6.6: Result of Proposed and Existing Models

Figure 6.6 shows the result of a comparison between the proposed keyframe-based saliency detection for human action recognition (KFSD HAR) and existing (CNN, VGG-LSTM, and VGG-16 + Bi-Directional LSTM) models. This result is obtained at training the network on the selected 20 class of dataset at 28th epochs. As shown from Figure 6.6, KFSD HAR outperforms the existing models by training and validation results. From the result, KFSD HAR was far better compared to CNN and VGG-LSTM on training accuracy and training loss, validation accuracy, and validation loss. Compared to KFSD HAR, VGG-16 + Bi-directional LSTM has nearly similar but lower results.

6.4 Discussion

In this study, the concepts of human action recognition were studied in line with the work of Amin Ullah.et al. c, which is based on using RGB data inputs to the network. So, this study is the modification of the stated work of Ullah.et al by the contribution of changing the preprocessing steps which include keyframe selection from video action clips and detecting salient features of keyframes. Then the input data was now the selected potential frame with its inclusive relevant information instead of using all frames per action clips and almost non-relevant features were included in the work of Ullah.et al. [56]. The idea of saliency detection for video action was taken from their recommendations and future directions, but the keyframe selection concept was an additional contribution made to this work.

Similar to Ullah.et al. [56], in this study, deep bi-directional LSTM which is an RNN network of two layers stacked on the top of each other was used. This enhances the recognition of frame to frame-sequential patterns of hidden states in the features. Therefore, the output of the frame at a time *t* is not only calculated from the previous frame at time *t-1* but also from the upcoming frame at time *t+1*. This is due to layers were performing forward and backward direction for learning sequential information. Finally, the combined output of both forward and backward layers was computed based on their hidden states.

Different from Ullah.et al. [56], there was a selected keyframe from action classes and performed a saliency detection procedures to include only relevant information while extracting features. Then preprocessed results were given to action recognition networks, i.e. bi-directional LSTM. For extracting frame features, VGG-16 was used. In this work, the UCF101 action dataset of 20 classes was selected for performing action recognition.

The dataset was divided by machine learning splitting protocol of 70% training and 30% for validation. For testing the performance, the accuracy of each action category was computed concerning its loss. The confusion matrix was used for performance evaluation regarding the accuracy of prediction. The evaluated result was presented using Figure 6.4. To state the negative outcome obtained, saliency detection does not provide an accurate and good result when the selected action has (1) uniform or nearly similar visualization to its scene, (2) non-uniform or irregular patterned action, (3) multi-actor per scene, and multi-action per frame and (4) improper position of detected action. On the other hand, in the action recognition network (bi-directional LSTM), some actions such as ApplyLipistic and JampingJack were

classified as JampingJack (1%) and HorseRiding (2%) respectively, due to (1) video level labels were used for frame-level labels. This video level labels were incomplete or even missing at frame or clip level that may lead false label assignment due to incomplete or missed information at video level labels, (2) detecting salient regions having similar visualization with its background has less result compared to expected detection and (3) inter-class similarity of action's motion.

Since action recognition takes input data from saliency results, poorly detected salient information of frames made less prediction of some action classes. However, the processing time is improved by 5 sec. (from 66sec. to 61sec.) compared to the VGG16+ Bidirectional LSTM model. This enhances fast processing time in overall training compared to the proposed work of Ullah.et al. [56]. The result of this distinction is found in Table 6.4. Additionally, the overall training accuracy was improved by 3.24 (from 82.724 to 85.964). Table 6.4 shows the result of achieved overall training accuracy. Moreover, the prediction result shows that the proposed KFSD HAR improves average prediction accuracy by 0.247 (from 0.75 to 0.997) compared to VGG-16-Bidirectional LSTM.

Generally, the proposed KFSD HAR model improves training time and accuracy as well as the prediction of actions. This is because keyframes from action video were selected and applied saliency detection on the selected keyframes to include informative features and remove irrelevant ones. Then the developed model for sequence learning used the result of the reconstructed video of salient keyframes. Compared to the existing VGG-LSTM, VGG 16, Bi-directional LSTM, and CNN model of human action recognitions, KFSD is fast for training the actions, produce better accuracy and loss at both training and validation as well as production.

6.5 Summary

In this chapter, a description of the obtained result by applying the developed model on the proposed work was given according to the process followed in each step. Concepts for gained result and how it be developed as described in section 5.1. The result provided by the experimentation stage and necessary steps were also represented in section 5.2.1. Next, evaluation of the result based on determined criteria which are the accuracy and loss of obtained results at training and testing were described in result evaluation section 5.2.2. Finally, the assessment of the result was performed using a defined confusion matrix.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORK

This chapter describes the conclusion, and research's future direction based on obtained results by implementing the developed keyframe saliency detection for human action recognition.

7.1 Conclusion

This study was proposed to developed deep learning-based human action recognition by using a key frame-based saliency detection algorithm for solving video action sequence problems like frame redundancy, the inclusion of irrelevant features, computational time. The study was attempted to develop and implement key frame-based saliency detection for human action recognition.

For the success of this study, procedures and stepwise tasks were conducted. Data collection from source, i.e. UCF101 and organized for use was done. Before directly using the data for the action recognition stage, two essential processes were conducted. These include first, keyframe selection process which is aimed at selecting frame(s) that contain potential information for representing the action, and second, saliency detection which is used for detecting salient information of selected keyframes from video clips. This step facilitates and simplifies the feature extra selection action process performed by the action recognition stage later on. By doing so, only potential frames with their relevant information were used for training action recognition networks, i.e. VGG16 + bi-directional LSTM for recognizing types of actions.

In this study, selecting data from the source of the University of Central Florida dataset UCF101 and organized into 101 classes was performed. This type of dataset was human action data in video format. For the first stage of this study, keyframe selection, only 20 classes of action data were selected since each class of action contains a range of 100 to 200 videos that require a long time while training proceeds. Then the selected actions of 20 class were segmented to frame sequences. Then computing their similarity based on RGB information and assigning index for each frame was performed.

Finally, computing the local maximum and comparing it with the segmented frame was done. Then frames with a greater or equal to an average of local maximum were stored as

keyframes in image formats. The parameters set at this stage were the frame's similarity measurement using Local Maximum and the maximum number of frames (50).

In the saliency detection stage, all the selected keyframes were given to the saliency detection network to detect only salient information of keyframes. The saliency detection network includes the pyramidal attention module for developing multi-attention features of frames, edge detection for detecting objects, and boundary for a clear separation between each other and the saliency module which develops salient information of objects processed in the attention and edge detection module.

The attended feature and detected edges were concatenated together and given to the ReLu activation function. Then the result of the concatenated two modules was again given to the saliency module for the final object saliency result using SoftMax.

By applying the developed keyframe-based saliency detection method, the research questions stated in section 1.4 were answered. The answer for RQ1 is that the top challenges identified in this study was frame redundancies, inclusiveness of irrelevant features such as background complexity, background clutter, illumination changes and illumination changes as studied in this proposed work.

Additionally, for RQ2, the problem of frame redundancy in the action was solved by selecting keyframes from video using developed keyframe selection method. Additionally, saliency detection solves the inclusiveness of irrelevant features while detecting frame features. So, detected salient features were used for fast feature extraction for action recognition.

Moreover, for RQ3, the application of keyframe-based saliency detection on human action recognition produces a fast process in training the network, a smaller number of frames were used for representing the action, and relevant features were detected which facilitates fast feature extraction in the recognition process.

Finally, recognition of action was performed by using bi-directional LSTM which also has feature selection using VGG-16. Then reconstructed video from sequences of saliently detected keyframe was given to the network and the procedures were done in forwarding and backward directional sequence learning.

By applying all the KFSD HAR, the overall training accuracy of 86% was achieved. Additionally, 61sec for the overall training time of the selected 20 classes using 28 was

achieved. Finally, an average of 99.7% prediction accuracy was achieved on the 5 classes of the selected dataset. From the comparison of proposed and existing model results, there is an improvement in training time and accuracy.

7.2 Future Work

In this study, a method for recognizing human action using detecting salient information from the selected key-frame was conducted. But the obtained result shows infancy stage performance that needs further study, especially for a real-time process. For instance, the key-frame selection process needs further considerations since such benchmarks were not available, validity may be lost. Additionally, it is implemented using computer vision algorithms in which some parameters were set manually by the user. Therefore, it is better in the future to study this challenge to get a valid result which is a full system based.

In the saliency detection process since the nature of the dataset is from complex action and the saliency model used was static object-based saliency instead of video, the detection quality was not as much expected for all selected actions. For instance, for multiple actors acting, no priority in the visualized scene, irregular patterned, and complex shapes. This is due to the developed saliency detection method focuses on the static saliency detection process, which needs additional dynamic salient information for real-time action recognition. Thus, static and temporal information would produce more accurate results when implemented for real-time processes. For the future, such tasks may be considered.

Generally, the proposed system was on the stage of improvement even if it is challenging to recognize complex action in real-world problems. In the future, further study should be given to overcome identified failures and challenges of this proposed work.

*This research was granted by Adama Science and Technology University with a grant number **ASTU/SM-R/088/19**.*

REFERENCES

[1] A. Kumar, “Panic Detection in Human Crowds using Sparse Coding,” MASc, University of Waterloo, Waterloo, 2012.

[2] Jemai Bornia, Sidi Ahmed Mahmoudi, Ali Frihida, Pierre Manneback, “Towards a Semantic Video Analysis using Deep Learning and Ontology,” *Assoc. Comput. Mach. New Delhi India*, vol. 1, Nov.2, 2018, doi: 10.1145/3241539.3241548.

[3] A. Bulling, U. Blanke, and B. Schiele, “A tutorial on human activity recognition using body-worn inertial sensors,” *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, Jan. 2014, doi: 10.1145/2499621.

[4] W. Jiang *et al.*, “Towards Environment Independent Device Free Human Activity Recognition,” in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking - MobiCom ’18*, New Delhi, India, 2018, pp. 289–304, doi: 10.1145/3241539.3241548.

[5] M. Hassaballah and K. Hosny, *Recent Advances in Computer Vision: Theories and Applications*. 2019, doi: 10.1007/978-3-030-03000-1.

[6] J. G. Shanahan and L. Dai, “Introduction to Computer Vision and Real Time Deep Learning-based Object Detection,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA, Aug. 2020, vol. 1, pp. 3523–3524, doi: 10.1145/3394486.3406713.

[7] S. Zhang, Z. Wei, J. Nie, L. Huang, S. Wang, and Z. Li, “A Review on Human Activity Recognition Using Vision-Based Method,” *J. Healthc. Eng.*, vol.1, 2017, doi: 10.1155/2017/3090343.

[8] M. Babiker, O. O. Khalifa, K. K. Htike, A. Hassan, and M. Zaharadeen, “Automated daily human activity recognition for video surveillance using neural network,” in *2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, Nov. 2017, pp. 1–5, doi: 10.1109/ICSIMA.2017.8312024.

[9] R. Khurana and A. K. S. Kushwaha, “Deep Learning Approaches for Human Activity Recognition in Video Surveillance - A Survey,” in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Jalandhar, India, Dec. 2018, pp. 542–544, doi: 10.1109/ICSCCC.2018.8703295.

[10] L. Wang *et al.*, “Temporal Segment Networks for Action Recognition in Videos,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2740–2755, Nov. 2019, doi: 10.1109/TPAMI.2018.2868668.

[11] F. Dirfaux, “Key frame selection to represent a video,” in *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, Vancouver, BC, Canada, 2000, pp. 275–278 vol.2, doi: 10.1109/ICIP.2000.899354.

[12] J. Li, Z. Liu, X. Zhang, O. Le Meur, and L. Shen, “Spatiotemporal saliency detection based on superpixel-level trajectory,” *Signal Process. Image Commun.*, vol. 38, pp. 100–114, Oct. 2015, doi: 10.1016/j.image.2015.04.014.

[13] A. Papazoglou and V. Ferrari, “Fast Object Segmentation in Unconstrained Video,” in *2013 IEEE International Conference on Computer Vision*, Sydney, Australia, Dec. 2013, pp. 1777–1784, doi: 10.1109/ICCV.2013.223.

[14] C. Guyon, T. Bouwmans, and E. Zahzah, “Robust Principal Component Analysis for Background Subtraction: Systematic Evaluation and Comparative Analysis,” in *Principal Component Analysis*, P. Sanguansat, Ed. InTech, 2012.

[15] Z. Tu *et al.*, “Fusing disparate object signatures for salient object detection in video,” *Pattern Recognit.*, vol. 72, pp. 285–299, Dec. 2017, doi: 10.1016/j.patcog.2017.07.028.

[16] “Video Background Subtraction in Complex Environments,” *J. Appl. Res. Technol.*, vol. 12, no. 3, pp. 527–537, Jun. 2014, doi: 10.1016/S1665-6423(14)71632-3.

[17] Yu Kong and Yun Fu, “Human Action Recognition and Prediction: A Survey,” *J. LATEX Cl. FILES IEEE*, vol. VOL. 13, no. NO. 9, Sep. 2018.

[18] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, “Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features,” *IEEE Access*, vol. 6, pp. 1155–1166, 2018, doi: 10.1109/ACCESS.2017.2778011.

[19] S. Kumari and S. Mitra, “Human action recognition using DFT,” Dec. 2011, doi: 10.1109/NCVPRIPG.2011.58.

[20] B. H. W. Guo, Y. Zou, and L. Chen, “A review of the applications of computer vision to construction health and safety,” 2018, Accessed: Sep. 29, 2020. doi: http://hdl.handle.net/2292/45351

[21] Q. Li, H. Cheng, Y. Zhou, and G. Huo, “Human Action Recognition Using Improved Salient Dense Trajectories,” *Comput. Intell. Neurosci.*, 2016, doi: 10.1155/2016/6750459.

[22] Hong-Bo Zhang 1,2,*, Yi-Xiang Zhang 1,2, Bineng Zhong 1,2, Qing Lei 1,2, Lijie Yang 1,2, “A Comprehensive Survey of Vision-Based Human Action Recognition Methods,” no. 25 February 2019, Feb. 2019, doi: 10.3390/s19051005.

[23] V. Megavannan, B. Agarwal, and R. V. Babu, “Human action recognition using depth maps,” in *2012 International Conference on Signal Processing and*

Communications (SPCOM), Bangalore, Karnataka, India, Jul. 2012, pp. 1–5, doi: 10.1109/SPCOM.2012.6290032.

[24] G. Guo and A. Lai, "A survey on still image based human action recognition," *Pattern Recognit.*, vol. 47, pp. 3343–3361, Oct. 2014, doi: 10.1016/j.patcog.2014.04.018.

[25] R. Khoshabeh and J. Hollan, "Spatio-temporal interest points for video analysis," Jan. 2009, pp. 3455–3460, doi: 10.1145/1520340.1520502.

[26] M. Li, H. Leung, and H. Shum, "Human action recognition via skeletal and depth based feature fusion," Oct. 2016, pp. 123–132, doi: 10.1145/2994258.2994268.

[27] H. Meng, N. Pears, and C. Bailey, "A Human Action Recognition System for Embedded Computer Vision Application," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, Jun. 2007, pp. 1–6, doi: 10.1109/CVPR.2007.383420.

[28] M.-L. Chiang, J.-K. Feng, W.-L. Zeng, C.-Y. Fang, and S.-W. Chen, "A Vision-Based Human Action Recognition System for Companion Robots and Human Interaction," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, Chengdu, China, Dec. 2018, pp. 1445–1452, doi: 10.1109/CompComm.2018.8780777.

[29] A. Bonner, "What is Deep Learning and How Does it Work?" *Medium*, Sep. 08, 2019. https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-f7d02aa9d477 (accessed Jan. 08, 2020).

[30] Di Wu, Nabin Sharma, and Michael Blumenstein, "Recent Advances in Video-Based Human Action Recognition using Deep Learning: A Review," *IEEE*, 2017.

[31] R. S. Ghewari, "Action Recognition from Videos using Deep Neural Networks," UC San Diego, 2017.

[32] "A new hybrid deep learning model for human action recognition," *J. King Saud Univ. - Comput. Inf. Sci.*, Sep. 2019, doi: 10.1016/j.jksuci.2019.09.004.

[33] "Action Detection Using Deep Neural Networks: Problems and Solutions - DZone AI," *dzone.com*. https://dzone.com/articles/action-detection-using-deep-neural-networks-proble (accessed Jan. 08, 2020).

[34] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential Deep Learning for Human Action Recognition," in *Human Behavior Unterstanding*, vol. 7065, A. A. Salah and B. Lepri, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 29–39.

[35] N. Jaouedi, N. Boujnah, and M. S. Bouhlel, “A new hybrid deep learning model for human action recognition,” *J. King Saud Univ. - Comput. Inf. Sci.*, Sep. 2019, doi: 10.1016/j.jksuci.2019.09.004.

[36] X. Yan, S. Z. Gilani, H. Qin, M. Feng, L. Zhang, and A. Mian, “Deep Keyframe Detection in Human Action Videos,” *ArXiv180410021 Cs*, Apr. 2018, Accessed: Nov. 27, 2019. [Online]. Available: http://arxiv.org/abs/1804.10021.

[37] “K. Simonyan and A. Zisserman, “Two-Stream Convolutional Networks for Action Recognition in Videos,” *arXiv:1406.2199 [cs]*, Nov. 2014, Accessed: Sep. 29, 2020. [Online]. Available: http://arxiv.org/abs/1406.2199.

[38] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, and J. Jiang, “A Simple Pooling-Based Design for Real-Time Salient Object Detection,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3912–3921, doi: 10.1109/CVPR.2019.00404.

[39] W. L *et al.*, “Temporal Segment Networks for Action Recognition in Videos.,” *IEEE Trans Pattern Anal Mach Intell*, vol. 41, no. 11, pp. 2740–2755, Sep. 2018, doi: 10.1109/tpami.2018.2868668.

[40] X. Yan, S. Z. Gilani, H. Qin, M. Feng, L. Zhang, and A. Mian, “Deep Keyframe Detection in Human Action Videos,” *ArXiv180410021 Cs*, Apr. 2018, Accessed: Sep. 20, 2020. [Online]. Available: http://arxiv.org/abs/1804.10021.

[41] T. Bouwmans and B. Garcia-Garcia, “Background Subtraction in Real Applications: Challenges, Current Models and Future Directions,” 2019.

[42] Q.-Q. Chen, F. Liu, X. Li, B.-D. Liu, and Y.-J. Zhang, “Saliency-context two-stream convnets for action recognition,” in *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, Sep. 2016, pp. 3076–3080, doi: 10.1109/ICIP.2016.7532925.

[43] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, “DeepVS: A Deep Learning Based Video Saliency Prediction Approach,” in *Computer Vision – ECCV 2018*, vol. 11218, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 625–642.

[44] A. Abdulmunem, “Human action recognition using saliency-based global and local features, ” 2017.

[45] Zengmin Xu, Ruimin Hu, Huafeng Chenb & Hongyang Li, “Action recognition by saliency-based dense sampling,” *Wuhan Univ. China © 2016 Elsevier BV*, Sep. 2016, [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2016.09.106.

[46] Shichao Zhang ,Enqing Chen ,L in Qi , Chengwu Liang, “Action Recognition Based on Sub-action Motion History Image and Static History Image,” *Sch. Inf. Eng. Zhengzhou Univ. Zhengzhou 450000 China*, ICCAE 201AD, doi: 10.1051/matecconf/2016560.

[47] Andargie Mekonnen, Dr. Subramanian Karpaga Selv2, Dr Vajeravel Sampath Kumar , Bisratie Tesfaye, “OPTICAL FLOW BASED ETHIOPIAN TRADITIONAL DANCE VIDEO CLASSIFICATION SYSTEM,” *GESJ Comput. Sci. Telecommun.,* 2016.

[48] Dr. KarpargaSelve Subramanian, Andargei Mekonnen (last), “Content Based Classification of Ethiopian Traditional Dance Videos Using Optical Flow and Histogram Of Oriented Gradient, vol. 6, no. 3 February 2016, pp. 371–380.

[49] Suge Dong, Daidi Hu , Ruijun Li and Mingtao Ge, “Human Action Recognition Based on Foreground Trajectory and Motion Di_erence Descriptors,” May 2019, doi: 10.3390/app9102126.

[50] F. Bashir and F. Porikli, “Performance Evaluation of Object Detection and Tracking Systems,” *IEEE Int. Workshop Perform. Eval. Track. Surveill. PETS*, Jan. 2006.

[51] C. Schwenke and A. Schering, “True Positives, True Negatives, False Positives, False Negatives,” in *Wiley Encyclopedia of Clinical Trials*, R. B. D’Agostino, L. Sullivan, and J. Massaro, Eds. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2007, p. eoct021.

[52] Nian Liu and Junwei Han. DHSNet:, “Deep hierarchical saliency network for salient object detection,” *CVPR*, pp. 2, 6, 7, 8, 2016.

[53] Pingping Zhang, Dong Wang, Huchuan Lu, Hongyu Wang, and Xiang Ruan. Amulet, “Aggregating multi-level convolutional features for salient object detection,” *ICCV*, p. 6,7,8, 2017.

[54] Qibin Hou, Ming-Ming Cheng, Xiaowei Hu, Ali Borji, and Zhuowen Tu, and Philip Torr, “Deeply supervised salient object detection with short connections,” *CVPR*, p. 1,2,7,8, 2017.

[55] W. Wang, S. Zhao, J. Shen, S. C. H. Hoi, and A. Borji, “Salient Object Detection With Pyramid Attention and Salient Edges,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 1448–1457, doi: 10.1109/CVPR.2019.00154.

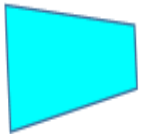
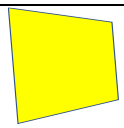
[56] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, “Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features,” *IEEE Access*, vol. 6, pp. 1155–1166, 2018, doi: 10.1109/ACCESS.2017.2778011.

[57] K. Soomro, A. Zamir, and M. Shah, “UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild,” *CoRR*, Dec. 2012.

APPENDIXES

A. Drawing Symbols Used and Their Representations

Table 0.1: Representation of symbols used in the drawing

No	***Symbols***	***Representation***	
1		Start / Stop	HAR Workflow Process Symbols
2		Input / Output	
3		Process	
4		Decision	
5		Workflow direction	
6		Saliency Detection Network	HAR Architecture Symbols
		Action Recognition Networks	
7		Convolutional Layer	
8		Pooling Layer	
9		Fully Connected Layers	
10		SoftMax Layer	

B. Sample of Codes

i. Keyframe Extraction

Load the video data, compute frame differences and select keyframes

```
cap = cv2.VideoCapture(str(videopath))
curr_frame = None, prev_frame = None, frame_diffs , frames =
[], []
success, frame = cap.read()
   if curr_frame is not None and prev_frame is not None:
      diff = cv2.absdiff(curr_frame, prev_frame)
      diff_sum = np.sum(diff)
      diff_sum_mean = diff_sum / (diff.shape[0] *
diff.shape[1])
      frame_diffs.append(diff_sum_mean)
      sm_diff_array = smooth (diff_array, len_window)
      frame_indexes =
np.asarray(argrelextrema(sm_diff_array, np.greater))[0]
```

ii. Saliency Detection

 # Load data, pretrained model and predict the saliency

```
testing_images = [datasest_path + '/images/' + f for dataset
in testing_datasets
  for f in os.listdir(datasest_path + '/images/')
     testing_images = [datasest_path + '/images/' + f for
dataset in testing_datasets
  for f in os.listdir(datasest_path + '/images/')
    m = Model (inputs=x, outputs = sam_vgg(x))
    m.load_weights('x.h5')
    predictions = m.predict(Ximg, batch_size = )
    res_saliency = postprocess_predictions(predictions
[9][0, :, :, 0], original_image.shape[0],
original_image.shape[1])
```

iii. Action recognition (Training and prediction)

```
from keras_video_classifier.library.recurrent_networks
import VGG16BidirectionalLSTMVideoClassifier
from keras_video_classifier.library.utility.plot_utils
import plot_and_save_history
from keras_video_classifier.library.utility.ucf.UCF101_loader
import load_ucf,scan_ucf_with_labels
load_ucf(input_dir_path)
classifier = VGG16BidirectionalLSTMVideoClassifier()

history = classifier.fit(data_dir_path=input_dir_path,
model_dir_path=output_dir_path, data_set_name=data_set_name)

if __name__ == '__main__':
    main()
```

#Prediction

```
load_ucf(data_dir_path)
predictor = VGG16BidirectionalLSTMVideoClassifier()
predictor.load_model(config_file_path, weight_file_path)
videos = scan_ucf_with_labels(data_dir_path, [label for
(label, label_index) in predictor.labels.items()])
video_file_path_list = np.array([file_path for file_path in
videos.keys()])
 np.random.shuffle(video_file_path_list)
 correct_count = 0, count = 0
 for video_file_path in video_file_path_list:
     label = videos[video_file_path]
     predicted_label = predictor.predict(video_file_path
     print('predicted: '+ predicted_label + ' actual: ' +
label)
     correct_count = correct_count + 1 if label ==
predicted_label else correct_count, count += 1, accuracy =
correct_count / count

if __name__ == '__main__':
    main ()
```