Summer 8-2021

Vision Based Activity Recognition Using Machine Learning and Deep Learning Architecture

Sarbagya Shakya

Recommended Citation

Shakya, Sarbagya, "Vision Based Activity Recognition Using Machine Learning and Deep Learning Architecture" (2021). *Dissertations*. 1926.
https://aquila.usm.edu/dissertations/1926

VISION BASED ACTIVITY RECOGNITION USING MACHINE LEARNING AND DEEP LEARNING ARCHITECTURE

by

Sarbagya Ratna Shakya

A Dissertation
Submitted to the Graduate School,
the College of Arts and Sciences
and the School of Computing Sciences and Computer Engineering
at The University of Southern Mississippi
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy

Approved by:

Dr. Zhaoxian Zhou, Committee Chair
Dr. Chaoyang Zhang
Dr. Sarah B Lee
Dr. Zhanxin Sha
Dr. Ras B. Pandey

August 2021

Published by the Graduate School

THE UNIVERSITY OF
SOUTHERN
MISSISSIPPI®

ABSTRACT

Human Activity recognition, with wide application in fields like video surveillance, sports, human interaction, elderly care has shown great influence in upbringing the standard of life of people. With the constant development of new architecture, models, and an increase in the computational capability of the system, the adoption of machine learning and deep learning for activity recognition has shown great improvement with high performance in recent years. My research goal in this thesis is to design and compare machine learning and deep learning models for activity recognition through videos collected from different media in the field of sports.

Human activity recognition (HAR) mostly is to recognize the action performed by a human through the data collected from different sources automatically. Based on the literature review, most data collected for analysis is based on time series data collected through different sensors and video-based data collected through the camera. So firstly, our research analyzes and compare different machine learning and deep learning architecture with sensor-based data collected from an accelerometer of a smartphone place at different position of the human body. Without any hand-crafted feature extraction methods, we found that deep learning architecture outperforms most of the machine learning architecture and the use of multiple sensors has higher accuracy than a dataset collected from a single sensor.

Secondly, as collecting data from sensors in real-time is not feasible in all the fields such as sports, we study the activity recognition by using the video dataset. For this, we used two state-of-the-art deep learning architectures previously trained on the big, annotated dataset using transfer learning methods for activity recognition in three different sports-related publicly available datasets.

Extending the study to the different activities performed on a single sport, and to avoid the current trend of using special cameras and expensive setup around the court for data collection, we developed our video dataset using sports coverage of basketball game broadcasted through broadcasting media. The detailed analysis and experiments based on different criteria such as range of shots taken, scoring activities is presented for 8 different activities using state-of-art deep learning architecture for video classification.

ACKNOWLEDGMENTS

Firstly, I would like to thank my primary supervisor and committee chair, Dr. Zhaoxian Zhou, for his continuous guidance, support, and mentorship throughout my academic career at the University of Southern Mississippi.

Secondly, I would like to thank Dr. Chaoyang Zhang, who also has provided remarkable influence, support, and guidance.

Also, I would like to thank my other committee members including, Dr. Zhanxin Sha, Dr. Sarah Lee, and Dr. Ras B. Panday for their valuable suggestions, feedbacks, and comments during the completion of my Ph.D.

I would like to thank Dr. Dia Ali, for his suggestions during the early part of the research and all the undergraduate students who have helped me in the data collection and preparation.

Also, I would like to thank all the faculty and staff of the School of Computing and Computing Engineering department for their support and family-like environment. Also, I would like to thank my colleague, lab mates, and friends with whom I got an opportunity to learn and collaborate not only in academic topics but in general topics. Their presence and guidance have motivated me a lot and helped me to gain knowledge. Ilemona Atawodi, Gabriel Idakwo, Haipeng Tang, Andrew Maxwell, Joseph Luitell, Trung T. Nguyen, Amartya Hatua, Shohel Rana Talukder, and Charan Gudla.

DEDICATION

This thesis is dedicated to my wife, Soneya Shakya, my parents, and all the family members whose constant support, encouragement, and patience towards me have driven me throughout this journey.

TABLE OF CONTENTS

ABSTRACT ii

ACKNOWLEDGMENTS iv

DEDICATION v

TABLE OF CONTENTS vi

LIST OF TABLES x

LIST OF ILLUSTRATIONS xi

LIST OF ABBREVIATIONS xvii

CHAPTER I - INTRODUCTION 1

0.1 Motivation 1

0.1.1 Background 3

1.2 Problem, Objective, and Contribution 9

1.3 Thesis Structure 10

1.4 Published paper related to this thesis 11

CHAPTER II BACKGROUND THEORY 12

2.1 Introduction 12

2.2 Model used 13

2.2.1 Machine Learning Architecture 14

2.2.2 Deep Learning Architecture 21

2.3 Datasets 30

2.3.1 Sensor-based dataset 31

2.3.2 Vision-based dataset 33

2.4 Evaluation metrics 39

2.5 Conclusion 42

CHAPTER III - SENSOR-BASED HUMAN ACTIVITY RECOGNITION 44

3.1 Motivation 44

3.2 Objective 45

3.3 Dataset Used 45

3.3.1 ACTi Tracker dataset (WISDM dataset) 46

3.3.2 Sensor Activity recognition Dataset (Shoaib SA) 46

3.4 Machine learning and deep learning architecture used. 47

3.4.1 Machine learning algorithms 47

3.4.2 Deep learning algorithms 48

3.5 Experimental results 50

3.6 Discussion and Conclusions 56

CHAPTER IV - VIDEO-BASED ACTIVITY RECOGNITION IN SPORTS 58

4.1 Motivation 58

4.2 Dataset Used 59

4.2.1 Olympic sports dataset 59

4.2.2 Sports video in the Wild (SVW) 60

4.2.3 UCF 101 sports categories 61

4.3 Deep learning architecture used. 62

4.3.1 C3D architecture 63

4.3.2 C3D + LSTM architecture 63

4.4 Methodology 64

4.5 Experimental Results 67

4.6 Discussion and Conclusions 71

CHAPTER V – ACTIVITY RECOGNITION IN BASKETBALL 72

5.1 Motivation 72

5.2 Literature review 73

5.3 Dataset Used 75

5.3.1 Data Collection 76

5.3.2 Dataset characteristics 80

5.4 Deep learning architecture used. 81

5.4.1 3D CNN 81

5.4.2 Two stream 3D CNN 82

5.5 Experiments 83

5.5.1 Subject dependent (SD) 84

5.5.2 Subject independent (SI) 94

5.6 Discussion and Conclusions ... 104

CHAPTER VI – FUTURE WORK ... 106

6.1 Future work ... 106

6.2 Limitations/Challenges ... 107

6.3 Conclusion ... 109

REFERENCES ... 111

LIST OF TABLES

Table 2.1 Sensor-based benchmark dataset 33

Table 2.2 Vision based benchmark dataset... 38

Table 3.1 Class distribution for WISDM dataset.. 46

Table 3.2 Summary of the CNN model layers, output shape, and a number of parameters. 49

Table 3.3 Comparison between two datasets used for the experiments. 50

Table 3.4 Accuracy of RNN and CNN architecture for dataset 1 and dataset 2.............. 54

Table 3.5 Precision and Recall value for all the activities for dataset 1 and dataset2 using CNN architecture. 55

Table 3.6 Accuracy Table using k-fold cross-validation 55

Table 4.1 Comparison of these three datasets.. 62

Table 4.2 Model parameter for C3D architecture. 68

Table 4.3 Result of 5-fold cross-validation for three datasets using C3D pretrained fine-tuning method. 68

Table 4.4 Accuracy table for all the models used for all three datasets........................... 70

Table 5.1 Accuracy for all the experiments with three models used. 104

LIST OF ILLUSTRATIONS

Figure 1.1 Types of human activity recognition based on input data 4

Figure 1.2 Categories and subcategories of HAR techniques... 7

Figure 2.1 Overall HAR process 13

Figure 2.2 Tree-like structure or Decision Tree.. 16

Figure 2.3 K nearest neighbor algorithm example for k =3.. 17

Figure 2.4 Tree-like structure of Random forest 19

Figure 2.5 Basic ANN Architecture.. 20

Figure 2.6 The architecture of the CNN network. 22

Figure 2.7 Generic Structure of an RNN model. 26

Figure 2.8 Single-cell of LSTM network... 27

Figure 2.9 The architecture of the LSTM model 28

Figure 2.10 Example of Two streams Convnet architecture [61] 29

Figure 2.11 Inflated Inception 3D architecture. Redrawn from[69] 30

Figure 2.12 The sample frame for different activities representing 4 modalities, RGB, 3D skeleton, depth, and infrared sequence (From left to right) data types from the NTU RGBd dataset. 36

Figure 2.13 Example of a confusion matrix for binary class classification...................... 41

Figure 3.1 Accuracy of three different machine learning algorithms for dataset 1 and dataset 2. 52

Figure 3.2 Precision, recall, and f1-score comparison of different activities for KNN architecture for dataset 1 and dataset 2 52

Figure 3.3 Accuracy and Loss curve of training data for CNN and RNN model (i) dataset 1 and (ii) dataset 2 53

Figure 3.4 Normalized confusion matric for predicted output in testing data with CNN model (i) dataset 1 (ii) dataset 2 53

Figure 4.1 Sample frame from 16 activities of Olympic sports dataset. 60

Figure 4.2 Sample frames of all 30 activities of SVW sports dataset. 61

Figure 4.3 Sample frame of all the activity in sports categories in UCF 101 dataset. 62

Figure 4.4 C3D Architecture.[119] 63

Figure 4.5 Proposed hybrid approach 63

Figure 4.6 Model architecture for C3D (left) and new model structure(right) 66

Figure 4.7 Accuracy and loss curve for training and validation data(left), a confusion matrix plot(right) for Olympic dataset with pretrained C3D+LSTM model 69

Figure 4.8 Accuracy and loss curve for training and validation data(left), confusion matrix plot(right) SVW sports dataset with pretrained C3D+LSTM model. 69

Figure 4.9 Accuracy and loss curve for training and validation data(left), confusion matrix plot(right) for UCF100 sports dataset with pretrained C3D+LSTM model. 70

Figure 5.1 Sample frame of optical flow video for different activity 78

Figure 5.2 An example frames of different scoring activities of the basketball dataset. .. 79

Figure 5.3 Class distribution of the dataset based on different groups 80

Figure 5.4 3D CNN network 82

Figure 5.5 Two stream CNN Network 82

Figure 5.6 The distribution of data to training and testing data for subject-dependent and subject-independent classification. 83

Figure 5.7 SD Confusion matrix plot and classification report for class-8 RGB (grayscale) input video.. 85

Figure 5.8 SD Confusion Matrix plot and classification report for class-8 RGB input video... 85

Figure 5.9 SD Confusion matrix plot and classification Report for class-4 RGB (grayscale) input video.. 86

Figure 5.10 SD Confusion Matrix plot and classification Report for class-4 RGB input video... 86

Figure 5.11 SD Confusion matrix plot and classification Report for class-2 RGB (grayscale) input video... 87

Figure 5.12 SD Confusion Matrix plot and classification Report for class-2 RGB input video... 87

Figure 5.13 SD Confusion matrix plot and classification Report for class-8 Optical flow (grayscale) input video.. 88

Figure 5.14 SD Confusion Matrix plot and classification Report for class-8 RGB Optical flow input video. 88

Figure 5.15 SD Confusion matrix plot and classification Report for class-4 Optical flow (grayscale) input video data 89

Figure 5.16 SD Confusion Matrix plot and classification Report for class-4 RGB Optical flow input video data 89

Figure 5.17 SD Confusion matrix plot and classification Report for class-2 Optical flow(grayscale) input video data.. 90

Figure 5.18 SD Confusion Matrix plot and classification Report for class-2 RGB Optical flow input video data 90

Figure 5.19 SD Confusion matrix plot and classification Report for class-8 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video 91

Figure 5.20 SD Confusion matrix plot and classification Report for class-8 two-stream 3D CNN network using RGB and Optical flow input video. 91

Figure 5.21 SD Confusion matrix plot and classification Report for class-4 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video 92

Figure 5.22 SD Confusion matrix plot and classification Report for class-4 two-stream 3D CNN networks using RGB and Optical flow input video... 92

Figure 5.23 SD Confusion matrix plot and classification Report for class-2 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video 93

Figure 5.24 SD Confusion matrix plot and classification Report for class-2 two-stream 3D CNN networks using RGB and Optical flow input video... 93

Figure 5.25 SI Confusion matrix plot and classification Report for class-8 RGB (grayscale) input video.. 94

Figure 5.26 SI Confusion matrix plot and classification Report for class-8 RGB input video... 94

Figure 5.27 SI Confusion matrix plot and classification Report for class-4 RGB (grayscale) input video.. 95

Figure 5.28 SI Confusion matrix plot and classification Report for class-4 RGB input video... 95

Figure 5.29 SI Confusion matrix plot and classification Report for class-2 RGB (grayscale) input video.. 96

Figure 5.30 SI Confusion matrix plot and classification Report for class-2 RGB input video.. 96

Figure 5.31 SI Confusion matrix plot and classification report for class-8 Optical flow(grayscale) input video 97

Figure 5.32 SI Confusion matrix plot and classification report for class-8 Optical flow input video 97

Figure 5.33 SI Confusion matrix plot and classification report for class-4 Optical flow(grayscale) input video 98

Figure 5.34 SI Confusion matrix plot and classification report for class-4 Optical flow input video 98

Figure 5.35 SI Confusion matrix plot and classification report for class-2 Optical flow(grayscale) input video 99

Figure 5.36 SI Confusion matrix plot and classification report for class-2 Optical flow RGB input video 99

Figure 5.37 SI Confusion matrix plot and classification report for class-8 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video. 100

Figure 5.38 SI Confusion matrix plot and classification report for class-8 two-stream 3D CNN networks using RGB and Optical flow input video... 100

Figure 5.39 SI Confusion matrix plot and classification report for class-4 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video. 101

Figure 5.40 SI Confusion matrix plot and classification report for class-4 two stream CNN networks using RGB and Optical flow input video... 101

Figure 5.41 SI Confusion matrix plot and classification report for class-2 two streams 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video. 102

Figure 5.42 SI Confusion matrix plot and classification report for class-2 two-stream 3D CNN networks using RGB and Optical flow input video... 102

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
HAR	Human Activity Recognition
ToF	Time of Flight
RBG	Red Green Blue
2D	2 Dimensional
3D	3 Dimensional
RFID	Radio-Frequency Identification
SDR	Software Defined Radio
HOG	Histogram of Oriented Gradient
MBH	Motion Boundary Histogram
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
KNN	K Nearest Neighbor
ANN	Artificial Neural Network
SVM	Support Vector Machine
LSTM	Long Short-Term Memory
GPU	Graphics processing units
CCTV	Closed Circuit Television
AOC	Area Unver Curve
TP	True Positive
TN	True Negative
FP	False Positive

FN	False Negative
RF	Random Forest
DT	Decision Tree
C3D	Convolutional 3D
I3D	Inflated 3D Convnet
SD	Subject Dependent
SI	Subject Independent

CHAPTER I - INTRODUCTION

1.1 **Motivation**

With the development in Artificial Intelligence (AI) and the computing power of a computer in recent times, activity recognition has seen a lot of progress in recent years. The main objective of human activity recognition (HAR) is to develop an automated system that can accurately determine the activity being performed by the human by analyzing various input data coming out from the different sources. The process may differ depending on the type of source, the input data being used, the architecture being developed to train the model, different activity categories, and the application field of the system. Although it has shown some real progress in this field, it is still not comparable with the recognition capability of a human. But it has shown very good application in real-life scenarios and has proved very beneficial to integrate technology to solve real-life problems in a different field, including but not limited to automated surveillance [1], healthcare [2], elderly care[3], sports [4][5], robotics [6], security[7], and broadcasting media [8].

Automated Video Surveillance: Automated video surveillance has been one of the areas where the application of human activity has shown great potential and requirements. For intelligent surveillance of crowd in areas like shopping malls, games, live concerts, streets, or monitoring high traffic and vehicles in highways, crossroads, traffic lights, parking lots, these systems have and can provide great ease to recognize unwanted and suspicious activities and tracking these individuals in the crowd. This can also be helpful to reduce reaction time, manage security personnel workload, and can automatically inform the concerned personnel about the situation to reduce security threats.

Healthcare: Regular monitoring of activities of patients can help health personnel to accurately recognize the medical condition, diagnose the patients' problems and work on the treatment of the patients[9]. For example, by analyzing the daily activities such as walking, running, bathing, cooking, exercise routine of the patients suffering from chronic diseases such as diabetes, cardiovascular, and obesity, elderly people[3][10], disabled or patients with physical disease such as Parkinson[11][12], and mental diseases such as depression, anxiety, hallucinations, memory problems, and dementia, the health personnel can have real-time information of the patients and will be able to manage and treat these patients in case of any sign of abnormal behavior or fall of a patient to the ground[13].

Sign language interpretation: One of the fields where Human activity recognition has been applied is for sign language recognition, interpretation, and translations to text or voice continuously and in real time[14]. The system that can automatically translate the sign language that uses hand gestures, hand kinematics, and facial expression for people to communicate with hearing-impaired people with higher accuracy will be a great contribution [15].

Sports: Among different areas of sports, a training-assisted system for monitoring player exercise, movement, fitness, injury prediction, and detection has been developed. Analyzing the movement of the player and identifying player action[16] during training in games has been provided greater inputs to enhance the performance of the player. Not only in real games, but the use of activity recognition has also had great application in online and video games such as online behavior change[17] can be detected for player modeling for tracking player's behavior.

Robotics: Robots that are equipped with the capability to recognize the activities performed by a human can be helpful to improve the daily life of people. An example is a domestic robot that can understand human activities and be able to respond to them accordingly can be useful for healthcare and elderly care[18].

Security: For effectively monitoring the security threats, traditionally human operators have to monitor the human activities from multiple camera views captured from the different cameras which can be stressful and inefficient. An automated system can be developed to detect different security-related activities and behaviors of the person such as fighting, aggressive behavior, and actions, carrying guns or explosives, etc. Also, these systems can be used to monitor individuals or groups of people for surveillance in other security-sensitive areas like banks, ATMs, airports[19], and metro stations.

Besides these, some of the other fields where the application of HAR has been popular in recent time are in Entertainment, for identifying actions in the movies, dance movement, smart homes, and education.

The opportunity of its application in a wide range of fields and as the research progress in this area it will provide a long benefit in different sectors to improve the quality of daily life of people in the community which has been a motivating factor to research this topic.

1.1.1 **Background**

Based on the source of data that is being used as input for activity recognition Human Activity Recognition can be differentiating in sensor-based activity recognition, vision-based activity recognition, and radio-based activity recognition as shown in figure 1.1.

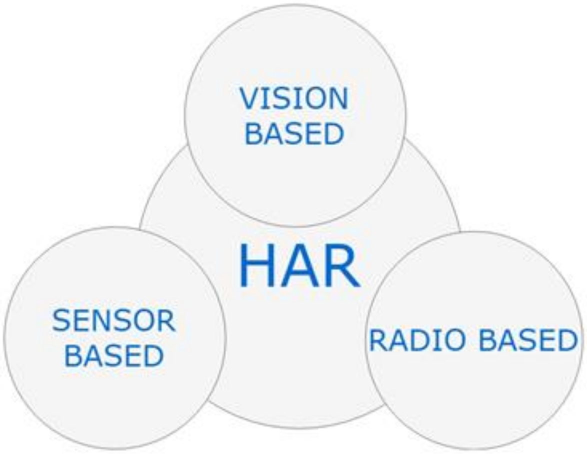


Figure 1.1 Types of human activity recognition based on input data

In **sensor-based** activity recognition, it will use the data collected from different inertial sensors such as accelerometer, gyroscope, and magnetometer that has been placed on different parts of the human body. This sensor detects the movements of the human and based on the data captured from these sensors, the activity performed by the person is classified.

With the recent development of wearable devices such as a smart watch, fit brit and smartphones equipped with power supply, memory, and sensors data communication capabilities have made it more feasible for data generation, data analyzing, and application in daily everyday life. Some of the disadvantages or challenges for sensor-based activity recognition can be the limitation in the battery life, size, inefficiency, unreliable and ineffectiveness of the sensor thus generating unstable data and inappropriate, uncomfortable, and confining person to wear sensor devices in different parts of the body for a longer period.

In **vision-based** activity recognition, the image/ video captured from different devices such as RGB camera, video recording devices, surveillance camera, or special design 3D camera such as 3D time of flight (ToF) camera, Microsoft Kinect camera, thermal cameras has been used as input data for activity recognition. Regular RGB camera provides 2D images or 3D videos whereas depth sensors such as a kinetic camera can provide depth images. The increase in the use of surveillance camera and video source platform like YouTube has made possible the availability of large visual data but some of the challenges it faced are a dependency of it on factors such as cluttered backgrounds, partial occlusion, viewpoint, different lighting conditions, appearance, camera angle, shadows in the image, and wide-angle low detail's view.

In **radio-based** activity recognition, it uses body attenuation and channel fading of the wireless radio signal to determine human gestures or activities[20]. Features such as signal attenuation, propagation, electromagnetic interference, fading characteristics as input data for activity recognition. Some of the radio types that are being used for activity recognition are ZigBee [21], Wi-Fi[22], RFID [23], radio waves, and software-defined radio(SDR)[24], etc. The main advantage of radio-based activity recognition is that it can be used in wide-scale applications, but factors such as reflection, refraction, diffraction, and interference due to the objects present in the environment can affect the overall accuracy of the system.

Besides sensor-based, vision-based, and radio-based activity recognition, a **multi-model** approach has also been used in recent times, where data from multiple sources is used for activity recognition. Like both visual data collected from the camera and sensor data collected from sensors at the same time are being used for training the model activity recognition. This approach has been useful in a different scenario where information obtained from only one approach is not sufficient or does not provide adequate information about the activity being performed. For example, the temperature, pressure exerted by the subject cannot be determined by only the video captured by the normal camera, where these values coming out from the sensor can help determine the activities performed by the person.

In our research, we have only considered the sensor-based and vision-based approaches and have only used data collected from sensors and cameras.

Based on different activities type performed by a human, the activity recognition can be group into several different categories. The classification of different categories of activities performed by people can be mainly classified as action-based, interaction-based, and motion-based.

In **action-based** classification techniques, movements conducted by a single person or group of persons such as walking, jogging, running, sitting, dancing, playing, etc. can be classified. Some actions also include making some postures or gestures or behavior changes with the ability to determine facial expressions that involve a specific body part such as hand or face without any verbal communication. Also, some actions that have some specific field applications such as fall detection which can relate to a patient that needs immediate response from the medical personnel, or ambient assisting living that can help in elderly care to assist people in their daily life. Hence this action-based can also be subcategories in gesture recognition, posture recognition, behavior recognition, activities of daily living recognition, fall detection and ambient assisted living.

Another category of Human activity recognition is **interaction based** in which activities involve the interaction of a human with an object or with another person. Human object interaction has wide application in fields like entertainment, robotics, human-computer interaction and so on that can use hand gesture or body movement to interact with machines or objects to perform specific tasks. Also, activities that include interaction between different people such as shaking hands, playing team games, or group activities fall into this category.

Another category of Human activity recognition can be defined as **motion-based** in which the motion of a human is used for activity recognition. This has applications in the field like surveillance, security in which it can use technology such as Wi-Fi RFID for motion sensing or also video surveillance for activity recognition. This can also be divided into three subcategories of tracking, motion detection, and people counting. Figure 1.2 [25] shows the detailed categories and subcategories of different activity recognition techniques.

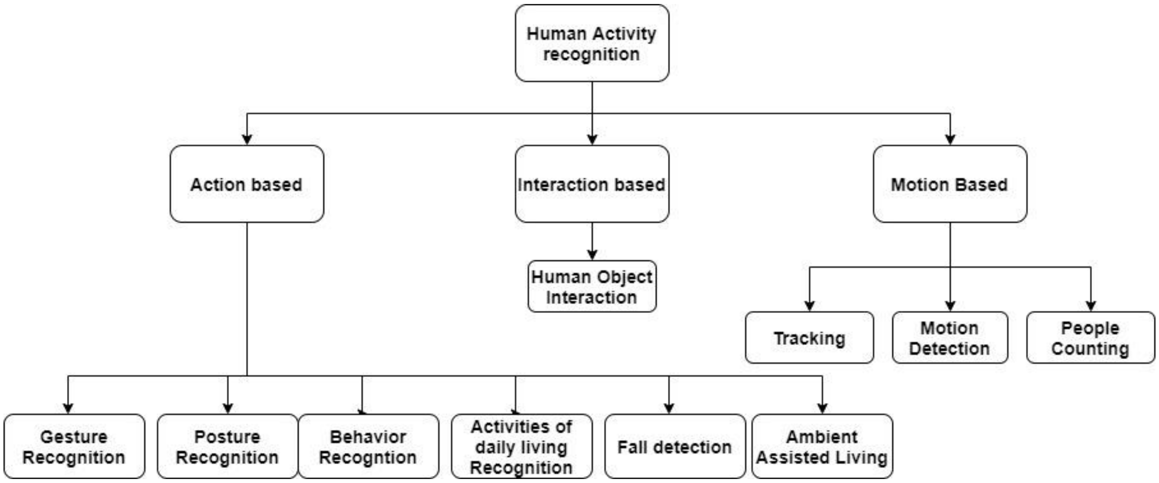


Figure 1.2 *Categories and subcategories of HAR techniques.*

Based on the feature representation, HAR can have three approaches [26]: model-driven approach, data-driven approach, and hybrid approach.

In a **model-driven** approach, activity models are produced using AI techniques such as rule-based systems, case-based reasoning, and ontological reasoning[27]. This approach is based on extracting hand-crafted features which are then used to classify activities. These features represent semantic concepts and their relationships based on prior knowledge such as histogram of oriented gradients(HOG), motion boundary histogram(MBH)[28][29]. But to generate hand-crafted features is time-consuming and requires a lot of inputs. Also, the extracted features could lack scalability and adaptability and cannot reflect all the information represented by the input data [30][31].

In a **data-driven** approach, the model is being trained with an existing large dataset. With the development of deep learning architecture in recent times, it has made it possible to replace the hand-crafted features with deep network features. These deep models can learn high-level intrinsic and abstract representation. But the requirement of large training data with specific parameters and training environment has made it inappropriate to all the fields. But recent advancements in the deep convolutional neural network, where the models such as AlexNet, VGG, GoogleNet, ResNet, etc that are trained in very large datasets such as ImageNet have been developed.

In **hybrid driven** approach, it combines the model-based and data-based approaches for activity recognition. In this approach, the model is provided with large input data for automatically evolving the model.

1.2 Problem, Objective, and Contribution

The challenges in the study vary with the types of input data, the difficulty level of activity, number of activities, and activity length [32]. This section describes the problems, the objective of the study, the contribution made in applying machine learning and deep learning architecture for activity recognition based on sensor-based and vision-based input data. The contributions are listed below:

In most of the sensor-based approaches, the problem will be to understand the number of sensors that can be placed on the human body to recognize the activity performed by that individual. As it will become agonizing for individuals to perform certain tasks carrying body-worn sensors. Therefore, a study was made to compare the change in the performance of the model using a single sensor with multiple sensors. Data collected from a single sensor(accelerometer) of a smartphone and multiple sensors on different parts of the body is compared for similar activities.

One of the major applications of vision-based activity recognition can be in the field of sports [33][34]. Up to now, a study has been made for the classification of different types of sports. Each player performing a certain activity in a sport is used as predicting information to recognize the sports. For example, throwing a ball, swimming, playing soccer, horse riding, etc. There was a lack of a specific dataset for a specific sport that can be used to classify different activities related to a single sport. Thus, we developed a dataset based on a basketball game that contains scoring activities related to basketball games from the video that has been broadcasted on the broadcasting media.

A preliminary evaluation of the developed basketball dataset is presented considering different scenarios to set the benchmark results and challenges the dataset brings in recognizing the activity related to basketball from the video broadcasted on live television. To our knowledge, this has been the first dataset of this type in which a labeled video dataset is prepared from the videos that can be used to classify the different scoring activities related to the basketball game. The future implementation of this project will help to design an automatic system in the field of sports such as automatic score updates, foul detection, assisting referees for decision making, and minimize human efforts for analyzing activities of players from the video replays. This research can be the initial contribution to develop an automated system recognizing human activity from a broadcasted video in real-time.

1.3 Thesis Structure

This thesis is constructed as follows. Chapter 1 outlines the motivation, introduction, and objective of the research. Chapter 2 describes the background theory of the overall HAR process, theoretical explanation of the machine learning and deep learning models that have been used throughout the experiments, Literature review of some of the benchmark datasets, evaluation metrics used to analyze the output performance. Chapter 3 provides the detailed study and comparison of the application of machine learning and deep learning architectures using the sensor data collected from sensors embedded in the mobile phone on different parts of the body. The chapter provides a detailed study, experiments, and results from the discussion for two publicly available datasets for activity recognition. The next chapter, Chapter 4 introduces the study of deep learning for activity recognition using visual data. The chapter also provides detailed studies, experiments, and result from the analysis of three publicly available benchmark datasets related to sports with future implementation. Next, chapter 5 introduces a new basketball dataset prepared for classifying different basketball-related scoring activities using different deep learning approaches. This chapter also provides a detailed study, experiments, and results of the dataset from different scenarios forming it into groups for analysis. Finally, chapter 6 provides the future works, limitations, and challenges, and a conclusion describing the contribution of the research in detail.

1.4 Published paper related to this thesis

Parts of this dissertation have been published as peer-reviewed journal publications, conference publications.

1. **Sarbagya Ratna Shakya**, Chaoyang Zhang, and Zhaoxian Zhou,” Basketball-51: A video Dataset for Activity Recognition in the Basketball Game, 2nd International Conference on Big Data, Machine learning and Applications, BIGML 2021, Vancouver, Canada.
2. **Sarbagya Ratna Shakya**, Chaoyang Zhang, and Zhaoxian Zhou, "Comparative Study of Machine Learning and Deep Learning Architecture for Human Activity Recognition Using Accelerometer Data," *International Journal of Machine Learning and Computing* vol. 8, no. 6, pp. 577-582, 2018.
3. Q. Liu, Z. Zhou, **S.R. Shakya**, P. Uduthalapally, M. Qiao, A.H. Sung,”Smartphone sensor-based activity recognition by using machine learning and deep learning algorithms”, International Journal of Machine Learning and Computing, 8 (2) (2018), pp. 121-126, 10.18178/ijmlc.2018.8.2.674
4. Z. Zhou, **S. Shakya** and Z. Sha, "Predicting countermovement jump heights by time domain frequency domain and machine learning algorithms", *Proc. 10th Int. Symp. Comput. Intell. Des. (ISCID)*, pp. 167-170, 2017.

CHAPTER II BACKGROUND THEORY

2.1 Introduction

Human activity recognition has been the problem of classifying the activities of a person analyzing the sequence of sensor data or visual data obtained from different types of input devices in real-time. With the development of self-sufficient gadgets and wearable devices in the form of a smartwatch, pulsometer, smartphone, and maximal use of surveillance cameras and phones equipped with a high-definition camera, the generation and collection of these data have been simple. Also, with the growth of the Internet of things and communication technology such as wireless data transmission, Bluetooth, or cellular data, these data can be easily transferred through a different medium and be used in modeling. This can be used to monitor the movement of the human without any delay in real-time.

Although it has been popular in the last decade, it still faces a lot of challenges to transform the received input data into known well-defined activity movements correctly and efficiently. In traditional methods, the time series sensor data from the sensors are used to generated hand-crafted features and used to train machine learning models such as decision trees, SVM, neural network, or ensemble of these models. In recent times with the development of deep learning and availability of large sensor data as well as visual data, models such as Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) have shown great promise and have provided state-of-the-art results for activity recognition. Figure 2.1 shows the overall HAR process which consists of different source for data generation such as different sensors, smartphone, camera, CCTV, the mode of transmission of data for analysis such as Bluetooth, training models, and architecture used and then classifying the activities on the testing data based on the trained model.

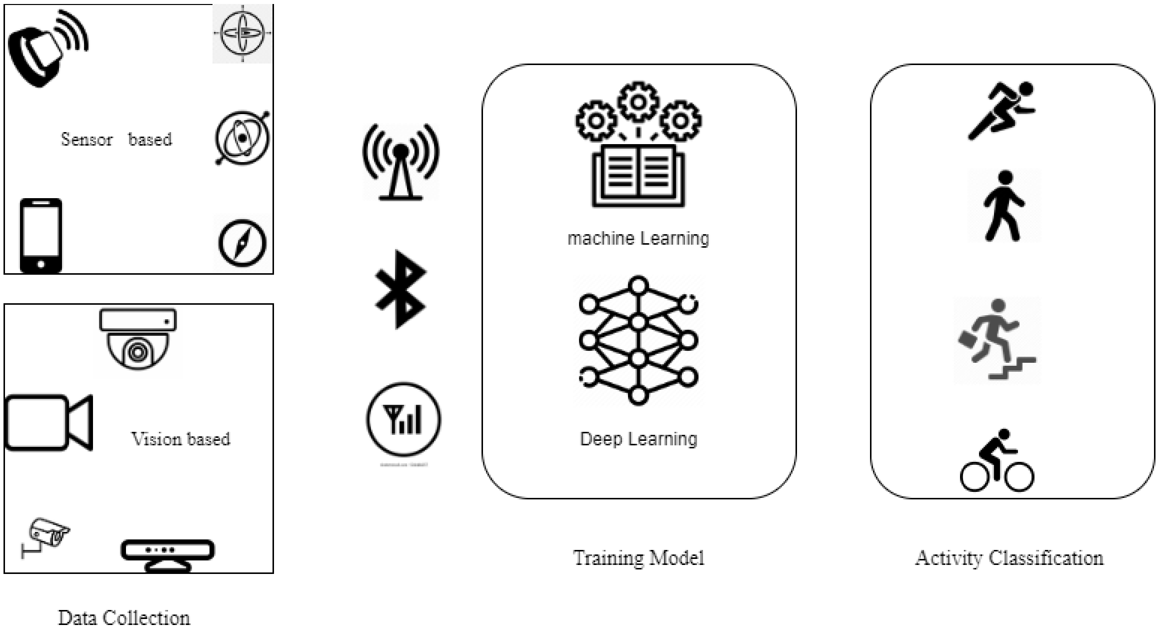


Figure 2.1 *Overall HAR process*

The following chapter presents the literature review of some of the state-of-the-art machine learning and deep learning models, architectures used for HAR. The existing sensor-based and vision-based dataset, and performance analysis measures for performance analysis is also explained in detail. Some of the models and datasets that have been used in our experiments have also been introduced in the latter part of the chapter.

2.2 Model used

Traditionally, machine learning architecture is used for activity recognition which uses hand-crafted feature extraction methods. Different features related to the input data from the different sources are crafted manually and use to optimize the weights of the machine learning algorithms. This will help to make the model optimize the weights and predict higher performance with the testing data. But lately, with the advancement of deep learning architecture along with the increase in the computational capacity of the system, the application of deep learning models has been used widely for different activity recognition. Although, the high computational requirement, the ability to learn useful features without any prior feature extraction methods makes it more suitable in real-world problems.

In this section, we will present background, technical details, and a review of some of the popular machine learning algorithms that have been used for human activity recognition. Also, in the next section, we will review some of the basics of deep learning architecture, its technical details, and why it is more suitable in our research will be discussed.

2.2.1 Machine Learning Architecture

Machine learning algorithms are used for extracting knowledge from the data. with different types of data and training type machine learning algorithms can be divided into four main categories: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. In a supervised learning algorithm, we train the model with the previously known examples, i.e., providing the algorithms with the desired outputs(labels) for the inputs and make the algorithms learn the pattern from them. The model is then used to produce desired output from an entirely unknown input. This process is also known as classification. In unsupervised learning, we have an unlabeled dataset that will have only the known input data. The algorithms will have no output data, so this technique is hard to understand and evaluate. This method is also known as clustering or visualization algorithms. In semi-supervised learning algorithms, it will have partially labeled and unlabeled data. It will use both the characteristics of supervised and unsupervised learning. Another category of machine learning is reinforcement learning, which is a reward-based technique where an agent can observe the environment, select, and perform the action and get rewards in return. It will help an artificial agent to define a policy based on its interaction with the environment and choose the action it needs to take based on that policy in that situation.

This section will introduce some of the basic machine learning architecture that has been used or is popular for human activity recognition.

2.2.1.1 Decision Tree

A decision tree[35] is a supervised machine learning algorithm that uses simple series of sequential decisions to reach a specific result. The sequential decision is dependent on the features and attributes of the training data. A decision tree consists of three components: nodes, edges, and leaf nodes. Nodes represent the tests or attributes at each stage; edges represent the answers to the node and the connection to the next node, Leaf node is the exit point of the decision tree. Figure 2.2 represents the tree-like structure of the decision tree. The circle in the figure represents the nodes. The arrow line represents the edges and the circles at the last layer represent the leaf node. While training, the algorithms will build the best way to construct the decision tree with the training data so that the testing data will reach the correct decision. Each branch will have the decision-making steps that lead to the required results. This algorithm is useful for small and nonlinear datasets. Application of decision tree algorithms is in many areas such as but not limited to engineering, law, and business.

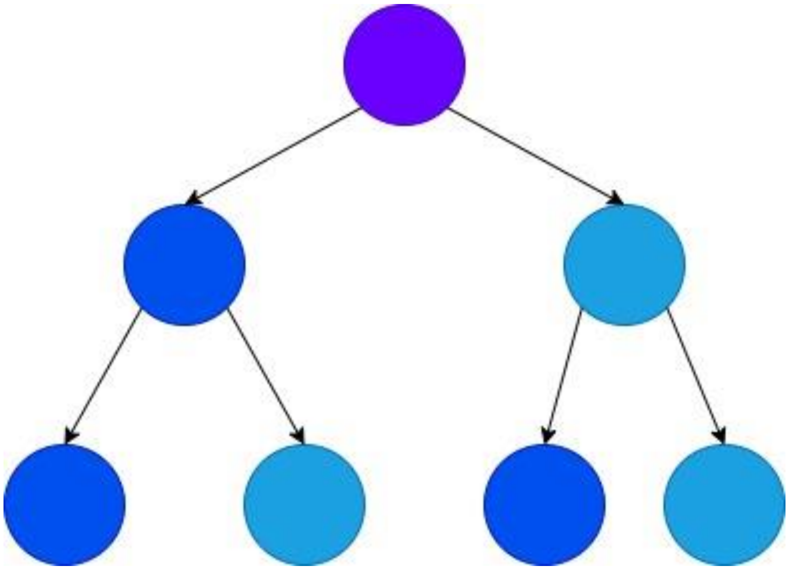


Figure 2.2 Tree-like structure or Decision Tree

2.2.1.2 K Nearest Neighbor

K-nearest neighbor (KNN) is the simplest Machine learning algorithm introduced first by Fix & Hodges in 1951[36] for performing patten classification tasks. KNN algorithm is based on supervised learning techniques where it stores the training data and finds the similarity between the new testing data with the available group of data. It puts the new data into the category that is most likely in the available categories. The KNN algorithms find similar features of that with the existing dataset and based on the most similar features it will classify as that output label.

Algorithms of KNN algorithms

Step 1: Define the number of the neighbors(k)

Step2: Calculate the Euclidean distance of k number of neighbors.

Step 3: Take the k nearest neighbors as per the calculated Euclidean distance.

Step 4: among these k neighbors, count the number of the data points in each category.

Step 5: assign the new data points to that category for which the number of the neighbor is maximum.

Step 6: model is ready.

Since in KNN it is required to calculate the distance between each data point, its computational cost can be high. Although, there is not any set of rules to determine the value of k at the beginning in general k is selected as k=n^ (1/2) where n is the number of data points. Selecting the value of k is important as a small value of k will have a higher influence of noise in the results whereas the higher value of k will increase the computational complexity. Figure 2.3 represents the k nearest neighbor algorithm example for k =3 where it has the similarity of 2 nearest neighbors in category 2 than in category 1. Hence it will be classified as category 2 labels.

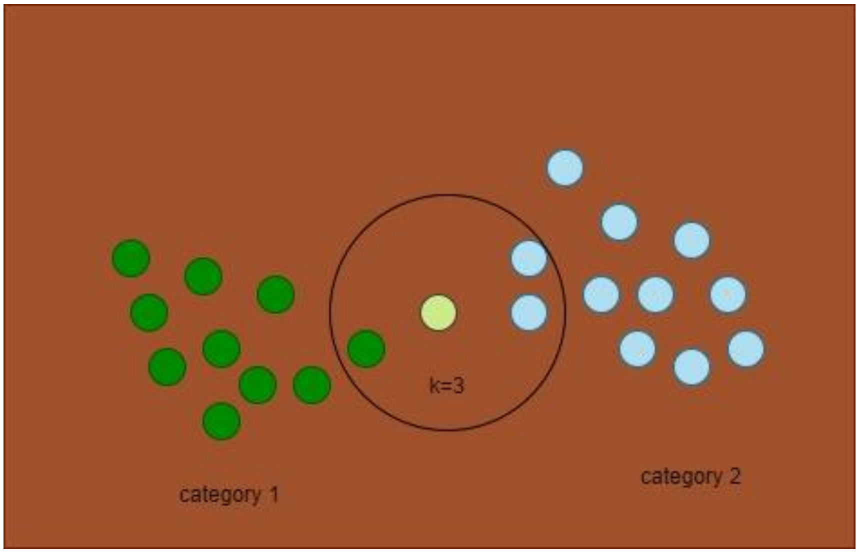


Figure 2.3 *K nearest neighbor algorithm example for k =3*

2.2.1.3 Random forest

Random forest[37] is also a tree-based machine learning algorithm that uses multiple decision trees for making decisions. It is a supervised algorithm that uses multiple trees for decision-making. The number of trees in the random forest affects the accuracy of the result. From the training data, the algorithm formulates some set of rules based on the input features and the target output and uses that same set of rules to predict the output to the testing data. A random forest can be used both for classification and regression. The algorithms create a node based on the randomly selected features from the input data and split the node into two best split daughter nodes. It will repeat the same process until the number of trees has been created. Once it builds the decision tree then it will use the rules to predict the outcome of the testing data. With output from each tree, it will calculate the votes for each predicted target and the highest vote predicted target will be the final prediction [38]. Figure 2.4 represents the tree-like diagram for the Random Forest with nodes 3. Hence it has three different decision trees where the output from each tree is used as a vote to select the majority voting to classify the final result. This algorithm is capable of handling large datasets, prevents overfitting issues, and has higher accuracy. Some of the sectors where Random Forest has been used are in sectors like banking, medicine, stock market, ecommerce, land use and marketing.

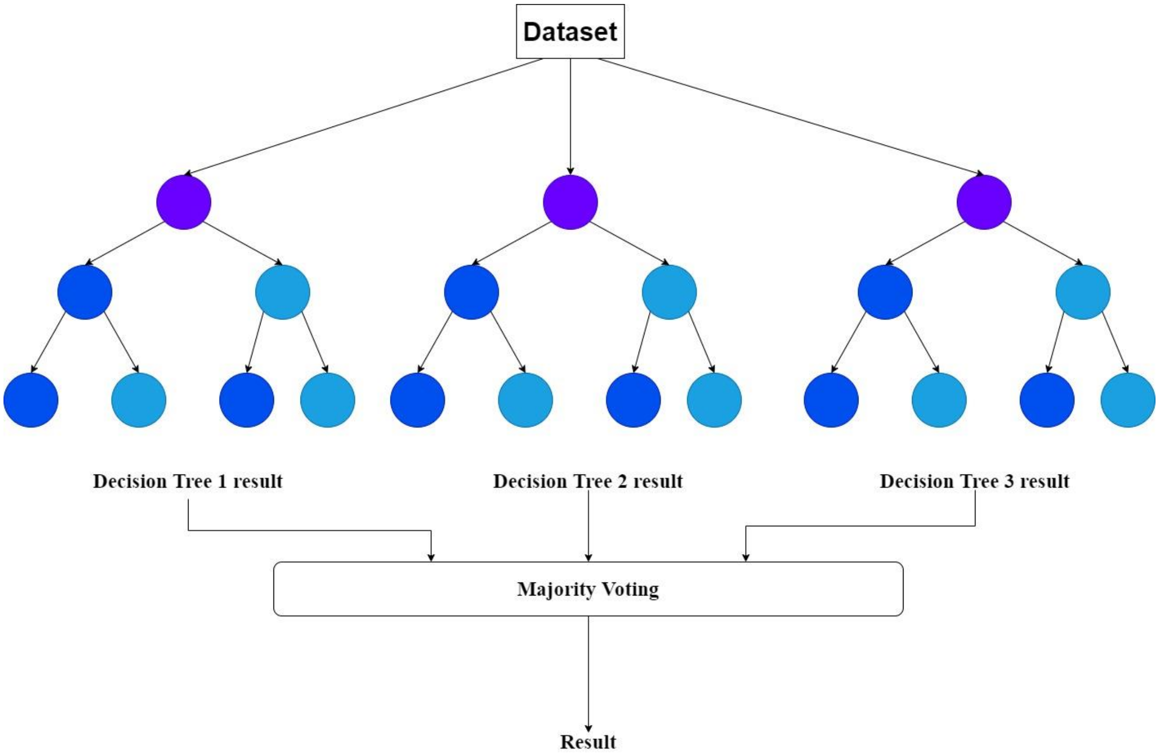


Figure 2.4 *Tree-like structure of Random forest*

2.2.1.4 Artificial Neural Network (ANN)

Artificial Neural Network is a special type of machine learning algorithm or a collection of algorithms that work similarly to the human brain. It consists of neural networks that can learn from the data provided to it for training and then predict or classify the output from the learned information. The ANN architecture discovers a pattern and relationship between the input and output data and was first introduced by Warren S McCulloch and Walter Pitts[39] in the 1970s. ANN consists of three layers as shown in figure 2.5: input layers, hidden layers, and output layer.

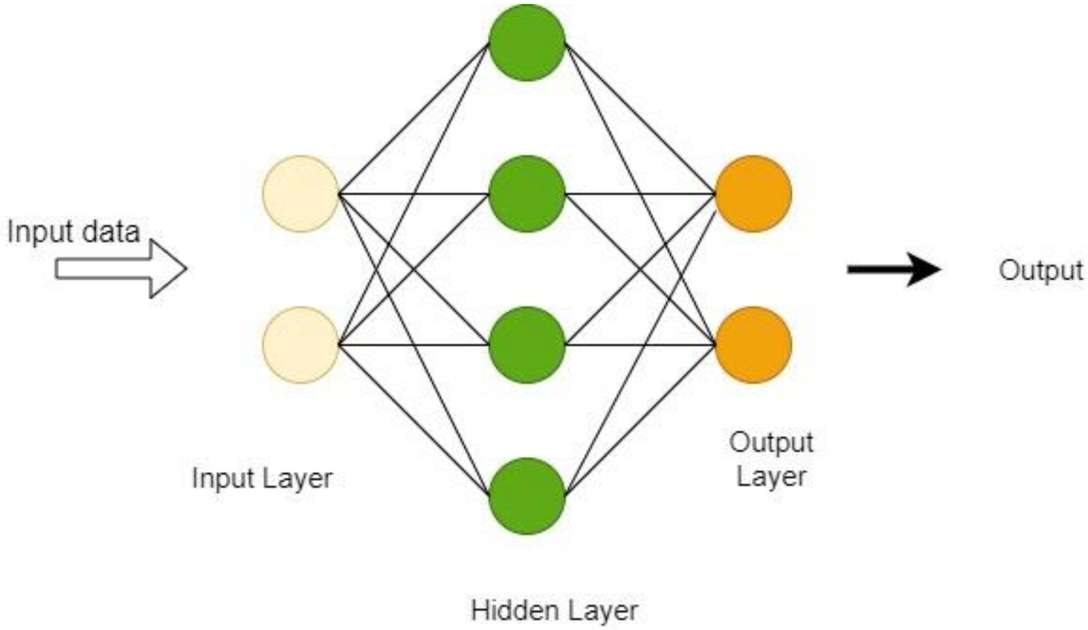


Figure 2.5 *Basic ANN Architecture*

Figure 2.5 represents the basic architecture of an ANN model with this three-layer. The input layer is the first layer that receives the input data for training or testing. The input data can be in any form in the form of numbers, text, images, videos, etc. The second layer is the hidden layer. There can be any number of hidden layers in the ANN network. This hidden layer performs various mathematical computations on the input data with its weights and bias value and generates features and recognition patterns from the input data. With the higher number of hidden layers, these have been categorized as deep learning which we will discuss in the next sections. The third layer is the output layer where we obtain the result or output after evaluating show correct the output is using various error functions. Deep learning, a part of ANN, has been widely used in different problem-solving fields such as handwritten character recognition, speech recognition, facial recognition, language translation, etc.

2.2.1.5 Support Vector Machine (SVM)

SVM algorithm[40][41] is one of the most powerful Machine learning supervised algorithms where the data is plotted in the n-dimensional space and an ideal hyperplane is defined to differentiate between the two classes. SVM can be used for both classification and regression problems. In SVM, the architecture finds the points that lie closest to both the classes which are known as support vectors. The distance between the points and the dividing line is the margin. The objective of SVM algorithms is to find the optimal line by maximizing the margin so that it reaches the maximum.

SVM has been used in many applications[42][16][43] such as face detection [44], text and hypertext categorization, bioinformatics[45], and handwriting recognition.

2.2.2 Deep Learning Architecture

Deep learning is the branch of Machine learning, where information from the data is processed through each layer which contains a uniform algorithm with one kind of activation function. In each layer, meaningful features of the data are constructed for training, learning, and understanding. The history of deep learning goes back to 1943 when Walter Pitts and warren Mcculloch [46] created computer model-based neural networks of the human brain. The development of continuous back propagation in 1960 by Henry J Kelley [47], models with polynomial activation functions in 1965 by Alexey Grigoryevich and Valentin were some of the significant developments. The first convolutional neural network develops by Kunihiko Fukushima in 1979[48], which uses an artificial neural network with a hierarchical, multilayered design called Neocognitron. In late 90's some significant development was made such as the development of support vector machines

(SVM)[40], Long short-term memory (LSTM)[49]. But, in early 2000 with the development of computational power of computer and Graphics processing units (GPU), the increasing computing speed helped to develop large models with higher efficiency and accuracy that could be trained with big data. In the next section, we will describe some of the basic state-of-art deep learning architectures.

2.2.2.1 Convolutional Neural Network

A convolutional neural network (CNN or ConvNet) is a part of Deep learning, which can recognize and classify features from an input image assigning learnable weights and biases and be able to differentiate one from the other. In recent years it has shown a high improvement in many fields such as image and video classification, computer vision image classification, facial recognition, image analysis, and natural language processing. One basic advantage of the CNN network is its ability to capture the spatial and temporal dependencies of the image and be able to reduce the image without losing features that help to design a more scalable model in the prediction process.

A basic CNN architecture consists of two parts. Feature extraction parts and classification parts as shown in figure 2.6.

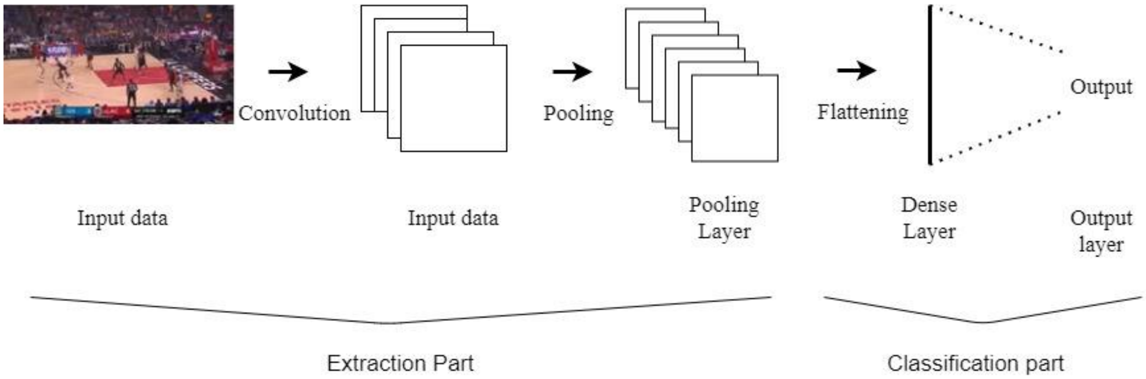


Figure 2.6 *The architecture of the CNN network.*

In feature extraction usually consists of three layers. Convolution layer and pooling layer The convolution layer is the first layer in CNN architecture. It extracts various features from the input image or video frames at different labels. In the convolution layer, the input image is convoluted with a filter by sliding the filter over the input image. This output from the convolution layer is called the feature map or the activation map. The convolution step can be 1D or 3D depending on the input image type. The initial layer extracts various low-level features of the image such as edge, corner color, gradient orientations, etc. This will be input to other higher label layers to extract more features from the input image. The higher label convolution layers extract high-level features which will give complete information about the input image.

The second layer is the pooling layer. This layer is a down sampling operation that decreases the spatial size of the feature map from the convolution layer by reducing the dimensionality of the connection between layers. This helps in reducing the computational power requirement. This layer also helps to extract dominant features from the feature map. Max pooling, average pooling, and sum pooling are some of the pooling methods that have been popular for CNN architecture. The CNN architecture consists of a number of convolutional and pooling layers depending on the complexity in the image to capture more features.

Fully connected layers are the layers in the classification parts where the features extracted from the feature extraction part are flattened and then used for predicting the output labels or class of the input image. This layer uses the weights and biases that connect the neurons to learn non-linear functions in that space. It is used to optimize objectives such as class scores. This layer is followed by the output layer which will classify the image using

SoftMax classification techniques. Besides these layers, another mostly used layer in CNN architecture is the dropout layer. This layer helps to reduce the overfitting problem by dropping some neurons from the network during training.

Many CNN architectures have been developed in a couple of decades that has provided remarkable achievement in the field of AI. The first one is LeNet-5[50] which was developed in 1998. Besides that some of the other milestone models and their architecture are Alex Net (2000)[51], Inception-V1(2006), Inception-V3(2008), ResNet-50(2011), Xception (2013), GoogleLeNet/Inception, Inception-V4(2015) [52] , Inception ResNets (2017)., ResNext-50(2019).

2.2.2.2 Recurrent Neural Network

The recurrent neural network is a class of neural networks that allows previous output to be used as input while having hidden states. Hence it is used mostly in time series sequential data such as speech recognition, natural language processing, text, speech recognition, and forecasting task. RNN can compute the current state from the current input and previous state output and finds the relationship between current inputs with the previously applied inputs. Hence it has at least one feedback connection so that the activation can flow in a loop. Figure 2.7 shows the generic unfold structure of an RNN model. The update rule of the RNN network can be defined as

$$a_t=b+W.h_{t-1}+U.x_t$$

$$h_t=\tanh(a_t)$$

$$O_t=c+V.h_t$$

$$y_t=SoftMax(o_t)$$

where x_t: is the input vector of input data or previous output data at time step t.

o_t is the intermediate output

h_t is an internal sate

U, V,W : weight parameters of the RNN model. These matrices are learnt by standard propagation.

y_t: output at time step t.

At each time step, the network receives as input $x_{t,}$ and then it emits an intermediate output o_t, in h_t internal state. Then this h_t and o_t will be feed to the next layer as a sequential input. One main drawback of the RNN model is the vanishing and exploding gradient problems[53] where the gradient values become zero and infinity as the gradient is multiplied together at each time step. During the gradient back-propagation phase, the gradient signal will be successively multiplied by the weight matrix many times. Because of this successive multiplication if the eigen value of the weight matrix is less than one then it will drive the gradient value to zero thus causing a vanishing gradient problem. And if the weight matrix is greater than one, then it will drive the value to infinity and cause exploding gradient problems. This problem is address by the Long short-term memory (LSTM) model with an introduction of a new structure called a memory cell. The details of the LSTM model are explained in the next paragraph.

Some of the popular applications of RNN in real life are in Siri, voice search, and Google translate.

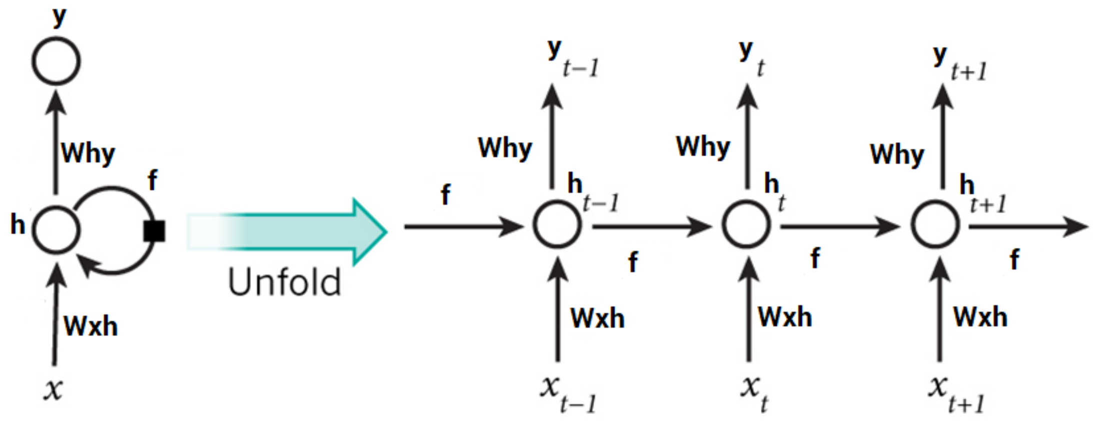


Figure 2.7 *Generic Structure of an RNN model.*

Long short-term memory (LSTM)

This is a type of RNN network first introduced by Sepp Hochreiter and Juergen Schmidhuber [49]in 1997. In the LSTM network, it has cells in the hidden layers which have three gates: input gate, output gate and a forget gate, and a neuron that connects itself and can store and delete data in the cell state. Figure 2.8 represents the architecture of a single cell of an LSTM network.

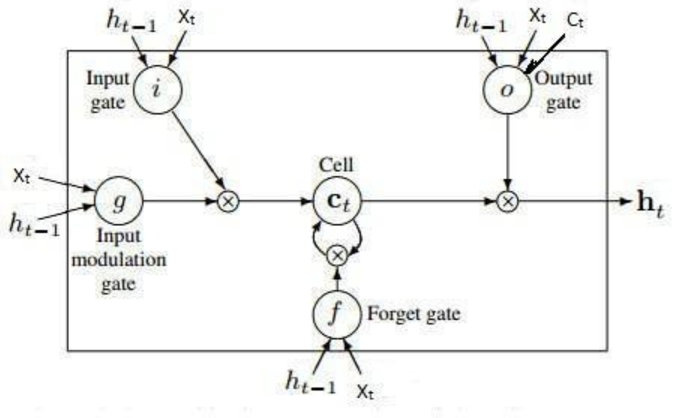


Figure 2.8 *Single-cell of LSTM network.*

It controls the flow of information that is needed to predict the output in the network. It is capable of learning long-term dependencies i.e. it can remember information for a longer period. Unlike the single neural network layer, tanh layer in most RNN networks, LSTM has four interacting layers as shown in figure 2.9. The update equation of the LSTM model is given by

$$i_t = g(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

$$f_t = g(W_{xi}x_t + W_{hi}h_{t-1} + b_f)$$

$$o_t = g(W_{xi}x_t + W_{hi}h_{t-1} + b_o)$$

$$\overline{ct} = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

$$c_t = f_t c_{t-1} + i_t\overline{c_t}$$

$$h_t = o_t \tanh(c_t)$$

where i_t: input gate

f_t: forget gate

o_t : output gate.

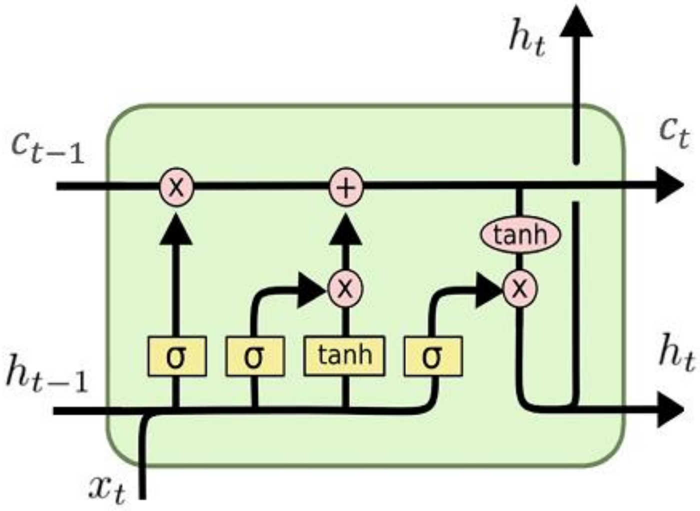


Figure 2.9 *The architecture of the LSTM model*

2.2.2.3 Hybrid Network

A hybrid network is designed by the fusion of two or more classifiers. It integrates two or more models, train them and combine their predictions for better result. This method is also called ensemble learning in Machine learning terms. This uses multiple models for the prediction of the same problems where the output from various models is average to predict the output[54]. Also, models such as CNN and LSTM are combined to learn both spatial and temporal features of the input data[55][56]. For example, for image classification, the CNN will generate the high-level spatial information of the activity of the images while the RNN model is used to extract temporal correlation between the consecutive frames of the clips by keeping the memory of the previous frames[57]. Besides ensemble of different models, hybrid approaches have been developed to combine shape based and motion-based features for representing an action. For this shape-based features are captured from the still image [58]whereas the motion features are captured using a histogram of motion intensity [59]or with the optical flow for action recognition. Also, two-stream networks in which one stream computes the information about the spatial information of the image whereas the other network computes the temporal information by capturing the displacement of optical flow from the video frames. As shown in figure 2.10 [60] described two-stream Convnet architecture where the spatial stream convNet is inputted with individual video frames whereas temporal stream convNet is given multiple frames of optical flow to capture the long context in the frames of the video.

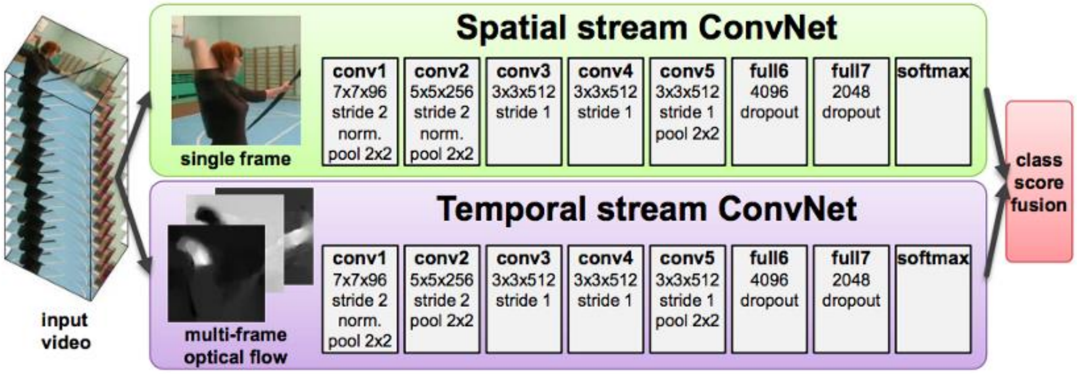


Figure 2.10 *Example of Two streams Convnet architecture* [61]

Also, there are different ways in where the two streams are concatenated. These approaches are named as late fusion[62][63], early fusion[64][65] or slow fusion and are implemented in different sectors[66] . Comparison between the performance of these different approaches has also been studied [67][68].

Besides that, another deep inflated model is developed such as the I3D model. The I3D model was first developed in DeepMind and the University of Oxford[69]. In this model, they add an additional dimension to the 2D architecture and inflates all the filter and pooling kernels. Inflating 2D convNets into 3D is the approach for video classification in which it converts 2D classification models into 3D by training multiple frames at once instead of one by one. Figure 2.11 shows the general architecture of the I3D model.

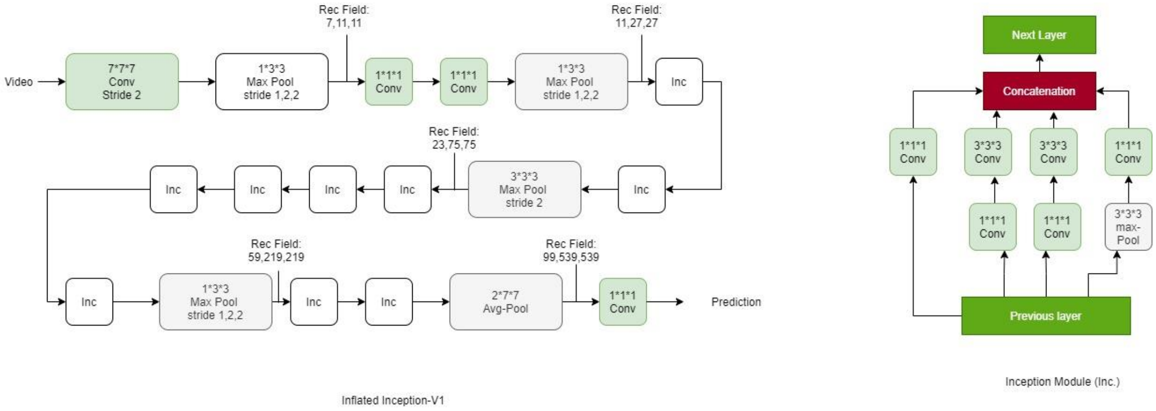


Figure 2.11 *Inflated Inception 3D architecture. Redrawn from*[69]

2.3 Datasets

Datasets are one of the most important parts of HAR research. The main objective of the HAR is to identify the action that is performed by the person from the data collected from a different source of data. Many studies have been carried out to manage this vast data and to relate these data to appropriate information or knowledge. For the study to be effective, a dataset that gives information on the actions under various conditions is required. Hence the availability and quality of the dataset hold crucial importance in training a model for recognizing activities accurately.

Many public datasets have been developed and published that have been used by many researchers to test their models and validate their proposals. The advantage of the publicly available dataset is that the comparison of different approaches and the capability can be measured if it is applied to the same datasets. Many datasets are recorded in controlled experimental environments with uniform backgrounds and static cameras which can be different from real-world scenarios.

Based on different factors, Datasets can be categorized into different groups such as based on source [70]: sensor-based or vision-based, based on viewpoint: single view or multi-viewpoint, based on actions performed by the subjects: action level datasets(A), behavior level dataset(B), interaction level dataset (I) and group activities label dataset(G).

This section describes some of the publicly available benchmark datasets that have been used for state-of-art human activity recognition in recent times. We have categorized the dataset into sensor-based and vision-based datasets along with its application in different modality and action types.

2.3.1 Sensor-based dataset

Most **sensor-based** datasets have data collected from body-worn sensors such as accelerometer, gyroscope, magnetometer, GPS, object sensors such as RFID tags, and ambient sensors such as radars, sound sensors, pressure sensors, and temperature sensors. Most body-worn sensors are embedded into different wearable devices such as a smartwatch, bracelets, or smartphones. These sensors detect the acceleration and angular velocity changed due to the human movement which is used for activity recognition[71][72]. Most body-worn sensors are used for activity recognition of daily living activities and sports. Object sensors are used mostly in smart home appliances [73]and medical activities[74] but it has been less applied than other sensors for HAR due to deployment difficulty.

Many sensor-based datasets have been published previously. Some of the benchmark datasets that have been published until now are explained in the next section.

UCI HAR dataset[75]**.** This dataset was published in 2013 and was constructed with experiments carried out for 30 subjects for six different activities such as walking, walking upstairs, walking downstairs, sitting standing, and laying. This dataset consists of a 3 axial linear accelerometer and 3-axial angular velocity captured from the embedded accelerometer and gyroscope of the smartphone. The time signals were sampled with a sliding window of 2.56s and a 50% overlap between them.

WISDM dataset [76]This dataset was published in 2011 and updated in 2013. This dataset consists of more than 2 million raw data collected from smartphone sensors for six different daily activities like walking, jogging, stairs, standing, and lying down. They have datasets collected through controlled laboratory environments as well as in the real world.

Opportunity dataset: This dataset [77]was developed in 2012 and was the subset of the Opportunity activity recognition dataset[78] acquired from 12 subjects while performing morning activities. The dataset was collected which includes 72 different sensors of 10 modalities integrated into the environment, in objects, and on the body in 15 wireless and wired networked sensor systems.

Some of the challenge's sensor-based dataset faced is the orientation of the sensors, types of sensors used, number of sensors used, the device orientation, the subject individuality, and body shape, and the number of activities needs to recognize from the data. Table 2.1 lists some of the benchmark's sensor-based dataset and its characteristics.

Table 2.1 *Sensor-based benchmark dataset*

Dataset`		Device	Sensors used	No of subjects	No of Activities	Variable-length segments
Du-MD[79]		Wearable	Accelerometer	33	7	Yes, manually
Ugulino et al[80]	2012	Wearable	Accelerometer	5	4	No
Opportunity dataset[77]	2012	wearable	accelerometers	4	35	No
USC_HAD[81]	2012	Wearable	Accelerometer	14	12	No
WISDM[76]	2012	Smartphone	Accelerometer	29	6	No
w-HAR[82]		wearable	IMU, stretch sensor	22	7	Yes
UCI HAR[75]	2013	Smartphone	Accelerometer	30	6	No
Shoaib et al[83]	2014	Smartphone	IMU	10	7	No
UniMiB SHAR[84]	2016	Smartphone	Accelerometer	30	9	No

2.3.2 Vision-based dataset

In a vision-based dataset, images and videos collected from different video capturing devices such as cameras, CCTV, or special cameras are used. In this section, we introduce some of the vision-based datasets that have been used.

KTH Activity Dataset.: This dataset[43] was first developed in 2004 which consists of 6 different activities performed by 25 subjects in four different controlled scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. The six activities, walking, jogging, running, boxing, hand waving, and hand clapping are performed by a single person using a static camera. It has different viewpoints and various scenarios and has activities related to action.

Weizmann Activity Dataset: This dataset [85]was first created in 2005 by the Weizmann Institute of science which consists of 10 natural actions performed by 10 subjects. The activities include running, walking, skipping, bending, jumping-jack, galloping-sideways, jumping-forward on two legs, jumping in place on two legs, waving two hands, and waving one hand which were recorded with a fixed camera with a simple background.

UCF sports: This dataset [86]was first developed in 2007 in the Computer Vision Lab, University of Central Florida. It includes 9 sports activities broadcasted on television. The activities include diving, golf swinging, kicking, lifting, horse riding, running, skating, swinging a basketball bat, and pole vaulting.

Hollywood Human activity dataset. This dataset [87]was developed in 2008 with 663 video samples obtained by 32 movie scenes labeled with at least one of eight actions: get out of the car, answer the phone, handshake, hug person, sit down, sit up, and kiss. It has two training sets of 223 samples in the automatic training set and 219 samples in the clean training set collected from 12 movies and 211 testing sets collected from 20 0ther movies.

Hollywood2 dataset.: This dataset[88] is an extended version of the Hollywood dataset published in 2009. It added 4 more classes action: run, driving the car, eat and fight in Hollywood dataset with a total of 12 class actions and adding samples for each class. The dataset consists of a total of 3669 video clips obtained from 69 movies.

HMDB51: This dataset[89] was published in 2011 by the Serre research lab at Brown University. This dataset consists of videos extracted from different sources such as google videos, YouTube. This dataset has 6849 clips divided into 51 different categories.

UCF101: This dataset was published in 2012 by the center for research in computer vision, University of Central Florida, USA. This dataset consists of 13320 videos collected from YouTube and is divided into 101 different categories. It is an extension of the UCF50 dataset.

Sports 1M: This dataset [90]consists of 1 million Youtube videos annotated with 487 sports-related classes where each class has around 1000-3000 videos per class.

Kinetics: This dataset[91] was initially published in 2017 by the DeepMind team which consists of YouTube video URLs. The initial Kinetics 400 contains 400 different human action classes and later been extended to Kinetics 600[92] and kinetics 700[93] with an increase I the class labels and video clips for each action class. This is one of the large-scale, high-quality datasets having a diverse range of human-focused actions.

Also based on different modalities [94], the Dataset can be categorized into four main groups: RGB, Skeleton(S), Depth (D), and Infrared(IR).

RGB modality refers to the images or videos captured from RGB cameras. Many benchmark dataset such as UCF101[95], HMDB51[89], Kinetics 400[91], kinetics 600[92] and kinetics 700[93] has been developed with RGB camera. The RGB data is easy to collect and has been applied in the field like sports, video surveillance but the large memory and computational cost requirements have made it more challenging in activity recognition.

Skeleton modality refers to the trajectories of human body joints, which characterize informative human motions. Mostly skeleton data is acquired by applying pose estimation algorithms on RGB videos[96] or depth maps[97] and motion-captured systems. The advantage of using skeleton data is its ability to give information and a simple representation of the pose and body structure of the subject in the image or video. Also, its robustness against the body structure, clothing textures, and background has made it more popular for activity recognition. Some benchmark dataset in that have skeleton modality includes CAD 60 [98], CAD 120[99], NTU RGB+D[100], and NTU RGB+D 120[101].

Depth modality includes a dataset of images where pixel value represents the distance information from a given viewpoint to the points in the scene. It provides 3D structural and geometric shape information of human subjects and is also used to convert 3D data into a 2D image. Sensors and special cameras such as time of flight, structured-based camera, stereo camera, Kinect Realsense3D are used to obtain depth images. The availability of low-cost and reliable sensors like Kinect has increased the use of depth data and videos more in HAR in recent years. Also, depth maps can be obtained from RGB videos using depth map estimation[102][103]. MSRDailyActivity3D[104], northwestern-UCLA[105] and UWA3D Multiview II dataset[106] are some of the benchmark datasets used for depth analysis.

Infrared Modality includes datasets that use infrared sensors, or thermal sensors to utilize target reflection rays to perceive objects in the scene or detect rays emitted from targets. This produces thermal images and videos which are used to extract spatial and temporal features used for activity recognition. An example of an infrared benchmark dataset is InfaR[107] which is mostly used in HAR experiments. Figure 2.11 shows the example of frames of different modality which are used for HAR.

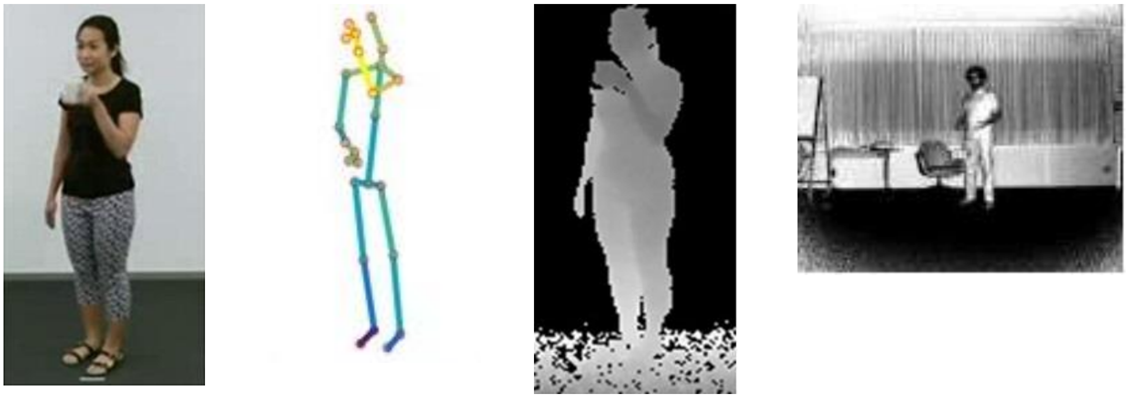


Figure 2.12 The sample frame for different activities representing 4 modalities, RGB, 3D skeleton, depth, and infrared sequence (From left to right) data types from the NTU RGBd dataset.

Most vision-based datasets have image and videos which are affected from different factors such as view changes, occlusion, light variation, variation in execution rate, anthropometry, camera motion, and background clutter and image/video quality. This has made these datasets very challenging for activity recognition. Table 2.2 list some of the vision-based benchmark dataset that has been released and used by most of the researcher for HAR.

Table 2.2 *Vision based benchmark dataset*

Dataset	Released year	# Of videos	# Of subjects	# Of class	Background	Modality	Activity Type
KTH[43]	2004	600	25	6	Clean static	RGB	A
Weizmann[85]	2005	81	9	10	Clean static	RGB	A
UCF Sports[86]	2008			10	Dynamic	RGB	
Hollywood[87]	2008	430	-	8	Dynamic	RGB	A, B, I, G
Hollywood2[88]	2009	1787		12	Dynamic	RGB	A, B, I,G
Olympic Sports[108]	2010	800		16	Dynamic	RGB	A, I
CAD 60[98]	2011	60	4	12	Static	RGB, S, D	
HMDB51[89]	2011	6766		51	Dynamic	RGB	A, B,I,G
MSRDaily Activity3D[104]	2012	320	10	16	Static	RGB, S, D	
UCF-101[95]	2012	13320		101	Dynamic	RGB	A, B,I,G
CAD 120[99]	2013	120	4	10	Static	RGB,S,D	
Thumos-2014	2014	18394		101	Dynamic		
Sports-1M[90]	2014	1133158		487	Dynamic	RGB	
Activity Net[109]	2015	27901		203	Dynamic		
NorthWestern-UCLA[105]	2014	1475	10	10	Static	RGB, S, D	
NTU RGB+D[100]	2016	56880	40	60	Static	RGB, S, D,IR	
YouTube 8M[110]	2016	8264650		4800		RGB	A, B,I,G
Something something v2							
Charades	2016	9848		157			
Kinetics 400[91]	2017	306245		400		RGB	
Something-something v1[111]	2017	108499		174		RGB	
AVA[112]	2017	437		80		RGB	
Kinetics 600[92]	2018	495547		600		RGB	A,B,I,G
Kinetics 700[93]	2019	650317		700		RGB	
Wang et al[113]	2019	1394	1	6		Wifi Csi	
NTU RGBD 120[101]	2019	114480	106	120	Static	RGB, S,D,IR	

S skeleton, D depth, IR Infrared

2.4 Evaluation metrics

In machine learning, to design an effective machine learning model evaluation metrics holds vital importance as it gives feedback to improve to get the desired performance accuracy. Different evaluation metrics are used for different problems. For different tasks like classification, regressing, ranking, clustering, topic modeling, etc. different evaluation metrics are used. The evaluation metrics not only gives the parameter to measure the performance of the model but also helps to explain the output from different implementation. Different evaluation metrics are being used to evaluate machine learning models' accuracy or performance. Some of them are confusion matrix, logarithmic loss, classification accuracy, precision, recall, the area under the curve (AOC), F1 score, mean absolute error, and mean squared error.

To define different performance metrics there are four important terms: True Positives (TP), True Negative (TN), False Positives (FP), False Negative (FN).

True positive is an outcome where the model correctly predicts the positive class. i.e., when we predicted Yes, and the actual output was also Yes.

True Negative is an outcome where the model correctly predicts the negative class. i.e., when we predicted NO and the actual output as also NO.

False Positive is an outcome where the model incorrectly predicts the positive class. i.e., when we predicted yes when actual output was No.

False Negative is an outcome where the model incorrectly predicts the negative class. i.e., when we predicted No when the actual output was Yes.

Below is the explanation of some of the popular evaluation metrics among which we have used throughout our experiments for analyzing the performance of your model in this research. As most of our problems are based on classification, the evaluation metrics are defined based on classification tasks.

Accuracy: Accuracy is a common evaluation metric for classification problems that gives the ratio of correct predictions to the total number of predictions made. It is mostly used for a balanced dataset having an equal number of samples for each class.

$$\text{Accuracy}=\frac{Number\ of\ correct\ prediction}{total\ number\ of\ predictions}$$

It can also be defined as

$$\text{Accuracy }=\frac{TP+TN}{TP+TN+FP+FN}$$

Precision: Precision is the number of true positives divided by the total number of elements labeled as the positive class. It summarizes the fraction of examples assigned to the positive class that belongs to the positive class. High precision means an algorithm returned more relevant results than irrelevant ones.

$$\text{Precision}=\frac{TP}{TP+FP}$$

Recall: Recall is the number of true positives divided by the total number of elements that belong to the positive class. It summarizes how well the positive class was predicted. it is like sensitivity. High recall means algorithms returned most of the relevant results.

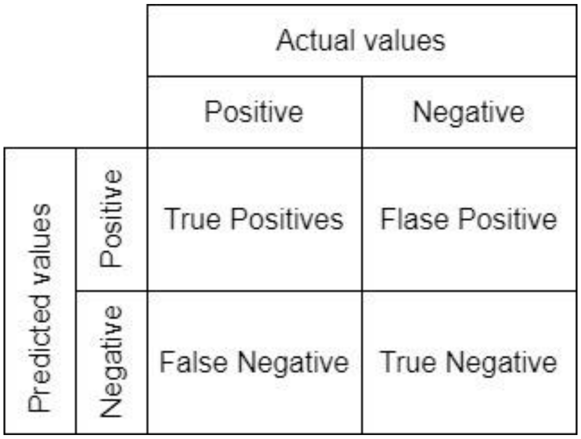
$$\text{Recall}=\frac{TP}{TP+FN}$$

F1-score: F1 score is used to measure the test's accuracy. The F1 score will give information about how precise and robust our classifier is. It is the harmonic mean of precision and recall. The range of the F1 score is from 0 to 1 where 1 represents the best value with perfect precision and recall and 0 represents the worst. It is mostly used to analyze the result with imbalanced classification.

$$\text{F1 score}=\frac{1}{\frac{1}{precision}+\frac{1}{recall}}$$

Confusion matrix: It gives the matrix of output and describes the complete performance of the model. It gives the tabular representation of the model predictions vs the actual values. Each row and column represent one class.

The binary confusion matrix can be represented as



		Actual values	
		Positive	Negative
Predicted values	Positive	True Positives	Flase Positive
	Negative	False Negative	True Negative

Figure 2.13 *Example of a confusion matrix for binary class classification*

Logarithmic Loss: It works by penalizing the false classifications. It is mostly used for multiclass classification. It measures the performance of a classification model where the prediction input is a probability value between 0 and 1. Log loss nearer to 0 means higher accuracy whereas loss away from 0 means lower accuracy. Hence its value will be in the range of [0, ∞)

$$\text{Logarithmic loss}=\frac{-1}{N}\sum_{i=1}^{N}\sum_{j-1}^{M}y_{ij}*\log\,(p_{ij})$$

Where N- number of samples

M number of classes

Y_{ij}=whether sample i belongs to class j or not

P_{ij} indicates the probability of sample I belonging to class j

Mean absolute Error: It gives the average of the difference between the original values and the predicted value. IT gives the measure of how far the predictions were from the actual output. It can be calculated by

$$\text{Mean Absolute error}=\frac{1}{N}\sum_{j=1}^{N}\left|y_j - \hat{y}_j\right|$$

Mean squared Error (MSE): MSE takes the average of the square of the difference between the original value and the predicted values. It is easier to compute the gradient in MSE. It can be calculated as

$$\text{Mean Square error}=\frac{1}{N}\sum_{j=1}^{N}\left(y_j - \hat{y}_j\right)^2$$

2.5 Conclusion

This chapter provides the background theory that is needed to conduct the human activity recognition and provided some of the literature reviews of the models that have been used in machine learning and deep learning architecture. Most machine learning architecture traditionally used hand-crafted feature extraction methods which with the development of deep learning architecture has made it easy as it generates features from the input data by itself.

This chapter provides the introduction of the benchmark dataset for activity recognition. As the dataset provides the higher contribution to the success of the model, it should provide the necessary information and should represent all the characteristics of the real-world environment. Based on different scenarios, these datasets can be grouped into different categories. We discussed some of the benchmark datasets based on this category which can help to under the problem better.

This chapter also describes some of the performance evaluation metrics that we have used throughout this research. Depending upon the nature of the problem and the output we are looking out for, the selection of the appropriate evaluation metrics is important to explain the outcome and evaluate the efficiency of the model. Some of the evaluation metrics are accuracy, precision, recall, F1 score, and confusion matrix to compare the output based on a different class.

CHAPTER III - SENSOR-BASED HUMAN ACTIVITY RECOGNITION

3.1 Motivation

With the increasing population of sensor technology, the sensor based HAR has also been popular and widely used in recent years. The sensor based HAR has motion data collected from smart sensors such as accelerometers, gyroscopes, Bluetooth, sound sensors, and so on. Also, the availability of these sensors inbuilt in commonly used devices like smartphones and more advanced wearable devices such as a wristband, smartwatches make it more appropriate to study the activities based on these sensors data. Some of the basic activities like walking, running, standing, sitting lying can be recognized using wearable accelerometers[114] which will measure human motion by measuring the linear 3D accelerations and orientations with respect to the earth's gravity[115]. Although it has shown some very good results using these wristband devices and smartwatches in the wrist and using multiple sensors placed on different parts of the body[13]. However, as a hand is our most active part, the data generated from these devices on the wrist of the person can be generated by the irregular movement which adds challenges to correctly recognize the activities based on these data. Also, the use of multiple sensors is uncomfortable and proved to be a burden to the person and is unfeasible in some applications such as in sports where the efficiency of players has been affected due to the body-worn devices during the game.

3.2 Objective

1. To compare the performance analysis of different ML models for the raw accelerometer data without any handcrafted data preprocessing and feature extraction method.

2. To compare the result between CNN and RNN model for time series data changing 1D sensor data to 2D data.

3. To compare the performance analysis of balanced and unbalanced data of similar nature with the same model and parameters.

4. To compare the performance analysis for data collected from a single sensor and multiple sensors.

3.3 Dataset Used

In our experiments, we have chosen two publicly available benchmark datasets: the ACTi tracker dataset commonly known as the WISDM dataset, and the sensor activity recognition dataset commonly known as the Shoaib SA dataset. Both of this dataset consists of data collected from accelerometer sensor of a smartphone placed in different body parts of the subject. Both datasets consist of different daily motion activities such as walking, sitting, standing, going upstairs, and going downstairs. The difference between these two datasets is in the WISDM dataset, only one accelerometer sensor is used which is carried in the waist by the subject whereas, the Shoaib SA dataset, used 5 accelerometer sensors carried in different parts of the body. Also, the number of data collected for the WISDM dataset is highly different for each class label whereas in the Shoaib dataset the amount of data for all the activities are exactly equal. Hence, the main objective of choosing these two datasets is to study and compare the results of two similar nature datasets but have a significant difference in the number of sensors used, the placement of the sensor, and to compare between the results for the balanced and highly unbalanced dataset. The details of these two datasets are described in the next section.

3.3.1 ACTi Tracker dataset (WISDM dataset)

This dataset[76] [116] was developed by the WISDM lab in 2013. It consists of raw data collected from the accelerometer of a smartphone attached to the waist of the volunteer performing six different activities in a controlled laboratory environment. The dataset consists of 2,980,765 labeled data. The different activities performed are walking, jogging, stairs, sitting, standing, and lying down collected from 29 users with a sampling rate of 20Hz i.e. One sample every 50 ms.

Table 3.1 *Class distribution for WISDM dataset*

Class Distribution:	Number	Percentage
Walking	1,255,923	42.1%
Jogging	438,871	14.7%
Stairs	57,425	1.9%
Sitting	663,706	22.3%
Standing	288,873	9.7%
Lying down	275,967	9.3%

3.3.2 Sensor Activity recognition Dataset (Shoaib SA)

This dataset[83] was collected in the university building by ten male participants aged between 25 and 30 years of age. The participants performed seven different physical activities for 3-4 minutes. The dataset consists of 630,000 data distributed equally among all the activity classes. The activities consist of walking, sitting, standing, jogging, biking, walking upstairs, and walking downstairs. The data were collected from 5 different accelerometers of a smartphone placed at five different body positions: right jeans pocket, left jeans pockets, right upper arm, right wrist, and on the belt position towards the right leg using a belt clipper. Although the original dataset consists of sensor data from accelerometers, gyroscope, and magnetometers, we have used only the accelerometer sensor value for experiments. All the activities have an equal number of sensor values which makes this dataset a balanced dataset.

3.4 Machine learning and deep learning architecture used.

For our analysis, we consider different machine learning and deep learning architecture. We tested many machine learning architectures with the raw data without any handcrafted feature extraction methods. Using only the raw data, we found some good results in some of the Machine learning architecture which is described in section 3.4.1. Also, for deep learning architecture, we used two models, CNN and RNN for activity classification. Although CNN has shown good results in image-based activity recognition and RNN is mainly used in time series data, we perform our experiments by changing our raw data in the form of time series. The detailed explanation of these architecture and experimental results are explained in the next sections.

3.4.1 Machine learning algorithms

For the experiments, we use different machine learning algorithms. Among them, the one which shows the highest performance are the K-nearest neighbor (KNN), Random Forest (RF) and Decision Tree (DT). For KNN we select the value of k as 5. For all the machine learning architecture we used the default hyperparameter using only the raw data collected from the sensors as input without any handcrafted feature extraction methods that are used traditionally for machine learning classification methods.

3.4.2 Deep learning algorithms

For analysis, we used two Deep learning architectures, 2D CNN and LSTM to develop a training model. The detail about this architecture is explained in the next section.

3.4.2.1 2D CNN

In our experiments, we have implemented 2D CNN for a time series 1D sensor data. 2D CNN is applied mostly in 2D image data and has performed a higher success in these fields. we tried to implement the same concept for the sensor data collected from the accelerometer of a smartphone. For that, we transform the 1D data to 2D data and reshaped the data to a 4D tensor to give it as an input to our 2D CNN model. The time-series data from the accelerometer is first divided into time series segments equal to the window size. We select the size of the window as 100 such that each segment will have a dimension of 1×100 and will be the same dimension as the 2D image. Here we overlap our segments with 50% overlapping techniques such that each segment will have half of the overlapping data from the previous segments. The depth axis gives the three-dimensional sensor values of the data. Our CNN model has four 2D convolution layers followed by a max-pooling layer, with a global average pooling layer followed by two dense layers. Two dropout layers with a probability constant of 0.5 are also in the model to reduce the overfitting of data. For training, the whole dataset was divided into 80/20 training and testing data. Also, 20% of the training data was further used as validation data. The model is trained with a

0.001 learning rate with the training data and the ReLU activation function was used in the training process. For each epoch, the CNN model was trained in the training data and then validate with the validation dataset. If the model improves in the validation loss, the model is saved. Hence at the end of the iteration. the model with the highest validation accuracy will be used to test the testing data. The analysis of the result is based on the performance of the model in the testing dataset. Table 3.2 shows different layers, output shape, and the number of parameters in each layer while training the CNN model used in the experiments.

Table 3.2 *Summary of the CNN model layers, output shape, and a number of parameters.*

layers	Output shape	# parameters
conv2d_1 (Conv2D)	(None, 1, 100, 16)	64
max_pooling2d_1	(None, 1, 50, 16)	0
conv2d_2 (Conv2D)	(None, 1, 50, 64)	1088
max_pooling2d_2	(None, 1, 25, 64)	0
dropout_1 (Dropout)	(None, 1, 25, 64)	0
conv2d_3 (Conv2D)	(None, 1, 14, 256)	196864
max_pooling2d_3	(None, 1, 7, 256)	0
conv2d_4 (Conv2D)	(None, 1, 7, 512)	131584
max_pooling2d_4	(None, 1, 4, 512)	0
dropout_2 (Dropout)	(None, 1, 4, 512)	0
global_average_pooling	(None, 512)	0
dense_1 (Dense)	(None, 50)	25650
dense_2 (Dense)	(None, #of class)	306

3.4.2.2 RNN

For our RNN model, we used the RNN model with LSTM cells. The model consists of three LSTM layers with a unit size of 100. The fixed training batch of size 25 was constructed with the training data. The training was one for 10 epochs with a learning rate of 0.3. After each epoch mean average and mean loss is calculated.

3.5 Experimental results

In our experiments, we have used two publicly available datasets of similar nature. Each dataset was prepared by collecting data from the tri-axial accelerometer sensor. But WISDM dataset (Dataset 1) have unequal distribution of data among its class whereas the sensor activity recognition dataset (dataset 2) has an equal amount of data among all the class. Also, the number of sensors used in dataset 2 is 5 whereas there is only one sensor used in dataset 1. The comparison between the two datasets is shown in table 3.3. Also, we have performed our experiments in most of the ML classifiers and have presented our results with the best three among them. These results are again compared with the DL algorithms.

Table 3.3 *Comparison between two datasets used for the experiments.*

Features	Dataset 1(WISDM)	Dataset 2(Sohail et dataset)
Number of examples	3005410	630000
Number of attributes/class:	6	7
Class Distribution	Walking: 1,255,923 (42.1%) Jogging: 438,871 (14.7%) Stairs: 57,425 (1.9%) Sitting: 663,706(22.3%) Standing: 288,873(9.7%) Lying Down: 275,967 (9.3%)	Walking: 90,000 (14.28 %) Sitting: 90,000 (14.28 %) Standing: 90,000(14.28 %) Jogging: 90,000(14.28 %) Biking: 90,000 (14.28 %) Walking Upstairs: 90,000 (14.28 %) Walking downstairs:90,000(14.28 %)
Nature	Unbalanced	Balanced
Number of sensors used	1	5
Sampling rate:	20Hz (1 sample every 50ms)	50 samples per sec
Sensor used	Accelerometer	Accelerometer

Figure 3.1 shows the overall accuracy of the three ML algorithms using raw data from dataset 1 and dataset 2. The model is trained with the training data and the trained model is tested with the testing data. From the figure among three ML classifier, KNN has the highest accuracy which is around 90% in dataset 1 and 97% in dataset 2 as compared to the other two ML classifier for both the dataset. Also, when looking at the performance

for each algorithm, accuracy is high in dataset 2 compared to dataset 1. The possible reason can be the balanced nature of dataset 2 where all the activities have an equal number of data whereas in dataset1 some of the activities like climbing stairs have only 1.9% of total data compared to walking which is around 42% of total data. Also, dataset 2 is collected from multiple sensors placed at different parts of the body. This should have provided more information for the model to learn and have enhanced the performance of the model. This shows that information from multiple sources or sensors will have a higher contribution to increasing the model performance. Also, to study the effect of the balanced nature of the dataset, we look at other performance metrics such as precision, recall, and F1 score. Since KNN classifier has the highest accuracy among all other ML classifier Figure 3.2 shows the comparison of these performance metrics for both the dataset. In dataset 1, precision, recall and F1-score is relatively high in all the activities except for climbing stairs. In dataset 2, precision, recall, and F1-score values for all the activities are high and consistent. This can be because of the balanced nature of dataset 2. Also, the similarity in the activities makes it hard to differentiate between the activities such as climbing stairs can be easily misclassified as walking as both are very much similar activities. In dataset 2 we see that the result is distributed along with all the activities. With uniform data distribution among all the classes the model will be able to generate more distinctive features among the different classes and will help to classify the activities more accurately.

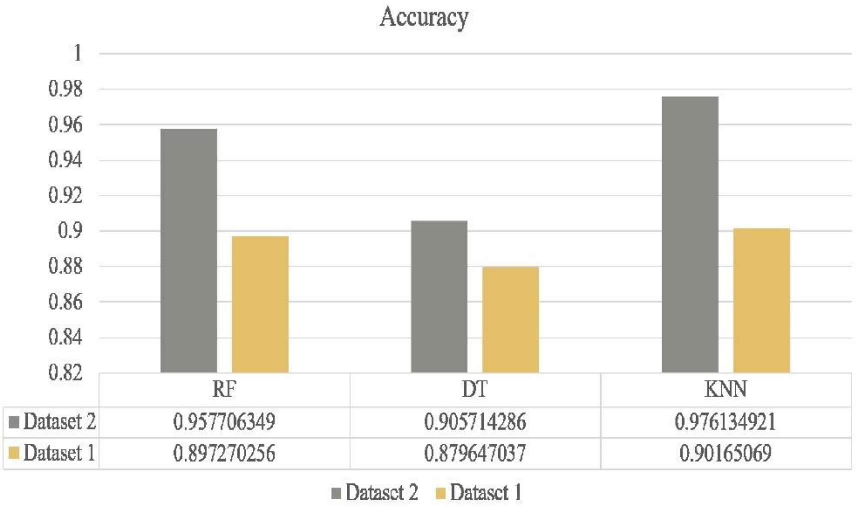


Figure 3.1 *Accuracy of three different machine learning algorithms for dataset 1 and dataset 2.*

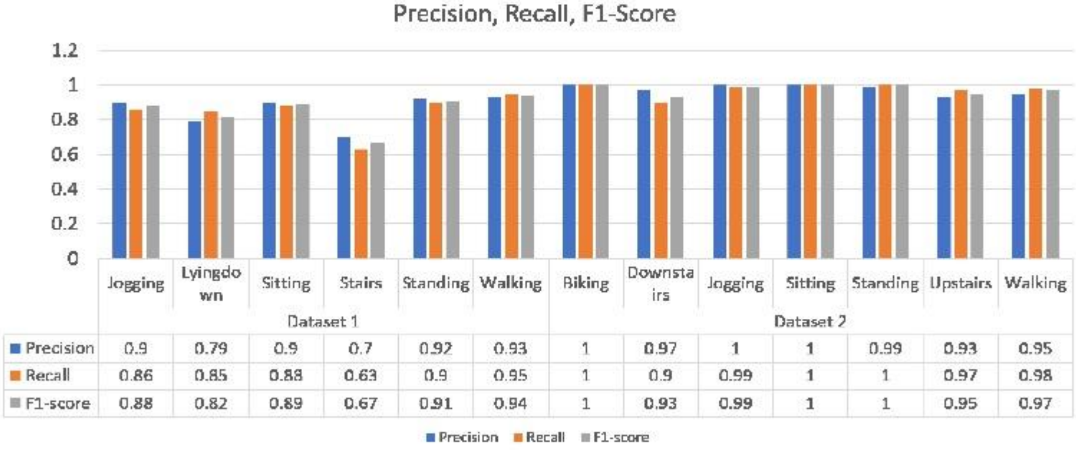
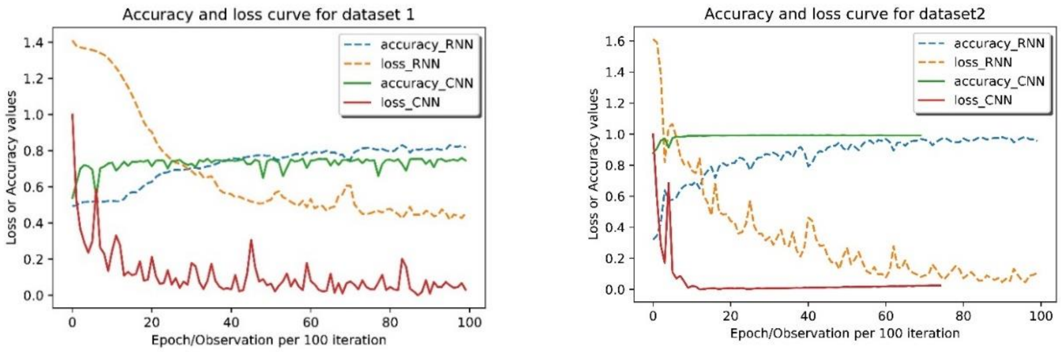


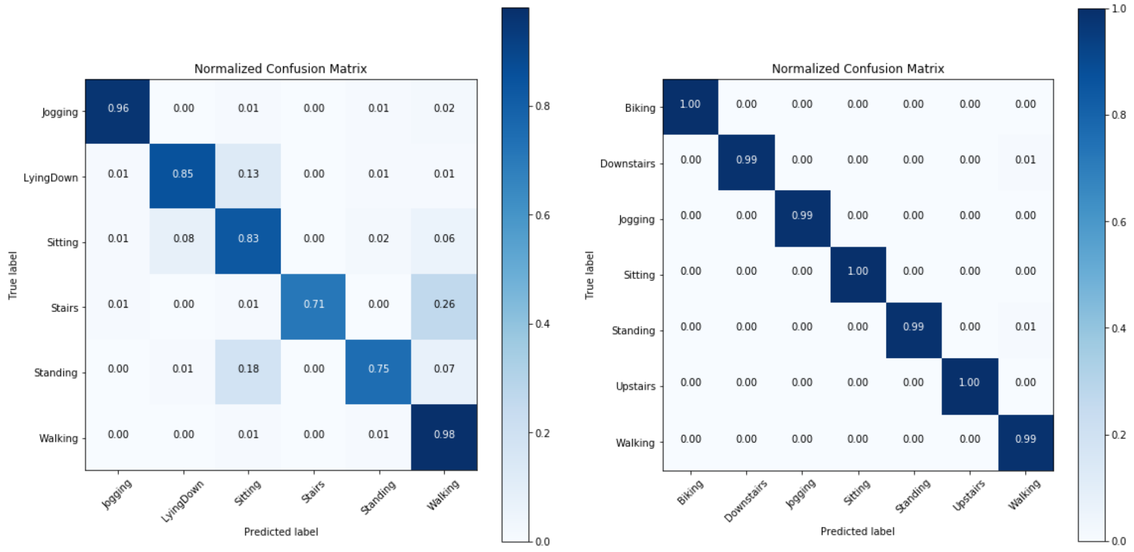
Figure 3.2 *Precision, recall, and f1-score comparison of different activities for KNN architecture for dataset 1 and dataset 2*

Thus, to compare the result coming from the ML classifier with deep learning architecture, we trained our model based on CNN and RNN. For that, we divide the whole dataset into training and testing datasets. Both the model is trained with the same set of training data and then tested with the testing dataset.



(i) (ii)

Figure 3.3 *Accuracy and Loss curve of training data for CNN and RNN model (i) dataset 1 and (ii) dataset 2*



	Jogging	LyingDown	Sitting	Stairs	Standing	Walking
Jogging	0.96	0.00	0.01	0.00	0.01	0.02
LyingDown	0.01	0.85	0.13	0.00	0.01	0.01
Sitting	0.01	0.08	0.83	0.00	0.02	0.06
Stairs	0.01	0.00	0.01	0.71	0.00	0.26
Standing	0.00	0.01	0.18	0.00	0.75	0.07
Walking	0.00	0.00	0.01	0.00	0.01	0.98

	Biking	Downstairs	Jogging	Sitting	Standing	Upstairs	Walking
Biking	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Downstairs	0.00	0.99	0.00	0.00	0.00	0.00	0.01
Jogging	0.00	0.00	0.99	0.00	0.00	0.00	0.00
Sitting	0.00	0.00	0.00	1.00	0.00	0.00	0.00
Standing	0.00	0.00	0.00	0.00	0.99	0.00	0.01
Upstairs	0.00	0.00	0.00	0.00	0.00	1.00	0.00
Walking	0.00	0.00	0.00	0.00	0.00	0.00	0.99

(i) (ii)

Figure 3.4 *Normalized confusion matric for predicted output in testing data with CNN model (i) dataset 1 (ii) dataset 2*

Figure 3.3 shows the accuracy and loss curve of the CNN and RNN model while training for both dataset 1 and dataset 2. In the curve, we can see that loss is decreasing and accuracy is increasing with each epoch which indicates the learning nature of the model. We did our experiment for 100 epochs/ iteration and shows the best result in both the dataset. The accuracy for each dataset for both the model is as shown in table 3.4. The accuracy for dataset 1 is around 81.74% with the RNN model whereas it is 92.22% in the CNN model. Also, the accuracy in the CNN model is high in dataset 2 which is around 99% than in dataset 1 which is around 96%. In both these datasets, using the CNN model has shown a better prediction than the RNN model for the time series sensor data.

When we look at the normalized confusion matrix for both of this dataset for CNN model output, as shown in figure 3.4, we can see that jogging and walking have been classified more accurately and the lowest recognition rate is of climbing stairs which have been mostly misclassified as walking and another one is sitting which has been mostly misclassified as standing. As both these pairs, walking-climbing stairs which are dynamic activities, and standing-sitting pair, which is a static activity, the similar nature of these activities can be the reason for the misclassification. Also, the insufficient training data in some of the activities such as climbing stairs which has very low data compared to other activities can be the reason for its low prediction rate. This imbalanced nature can be the reason for the misclassification and low prediction output.

Table 3.4 *Accuracy of RNN and CNN architecture for dataset 1 and dataset 2.*

Method	Dataset 1	Dataset 2
RNN	81.74%	95.65%
CNN	92.22%	99.12%

Table 3.5 *Precision and Recall value for all the activities for dataset 1 and dataset2 using CNN architecture.*

Dataset1			Dataset 2		
Activity	Precision	Recall	Activity	Precision	Recall
Jogging	0.97	0.96	Biking	1	1
Lying Down	0.80	0.85	Downstairs	0.99	0.99
Sitting	0.84	0.83	Jogging	0.99	0.99
Stairs	0.97	0.71	Sitting	1	1
Standing	0.89	0.75	Standing	1	0.99
Walking	0.93	0.98	Upstairs	1	1
			Walking	0.98	0.99
Avg/total	0.90	0.90		1	1

The precision and recall value for different activities is as shown in table 3.5. Here we can see that dataset 2 has better performance for all the activities than for dataset1. This also provides evidence that the use of multiple sensors and a more balanced dataset can provide higher recognition accuracy.

Table 3.6 *Accuracy Table using k-fold cross-validation*

	KNN		CNN	
Fold	Dataset 1	Dataset 2	Dataset 1	Dataset 2
1	90.15%	97.62%	84.37%	99.52%
2	90.23 %	97.64%	88.26%	99.52%
3	90.19%	97.68%	86.39%	98.41%
4	90.16%	97.67%	89.11%	99.36%
5	90.24%	97.63%	87.39%	98.96%
Average	90.19%	97.65%	87.31%	99.16%

To further validate our result, we analyze the result using 5-fold stratified cross-validation for the ML and DL architecture which has shown the highest performance among other architecture. Table 3.6 shows the accuracy for each fold and overall average accuracy for both the dataset. We can see that the result is consistent for all the fold. The average accuracy is higher in Dataset 2 in both cases. Comparing the KNN and CNN architecture we can see that DL architecture has higher accuracy in dataset 2 whereas in dataset 1 KNN has higher accuracy than compared with the CNN architecture.

3.6 Discussion and Conclusions

In this section, we implement ML and DL architecture to train a model for activity recognition using two datasets that consists of sensor data collected from different sensors of the smartphone. Most of the sensors that are used are accelerometers, gyroscopes, and magnetometers, but in our experiment, we used only the data collected from the accelerometer. The two-dataset used have similar nature daily life activities such as walking standing, climbing stairs, lying down, jogging and biking. The difference between these two datasets has been the number of sensors used and the nature of the dataset. Also, to make our experiment more real-life compatible we used only the raw data for the experimental setup without any hand-crafted features extraction. Among all the ML classifiers used KNN has shown the highest performance whereas in DL architecture used CNN has shown higher performance than the RNN model. In all the cases, dataset 2 which is a balanced dataset has outperformed the result compared to the unbalanced dataset 1. Most of the misclassification is of similar nature activities and the activity which has very little training data has been responsible for reducing the overall performance of the model.

This shows the overall comparison of ML and DL architecture in between two similar nature datasets with slightly different data distribution and sensor number. Techniques such as resampling where we can either oversample the class with very little data or down sample the class with very high data can be done to make a more balanced dataset. Besides some ensemble models and hybrid models can be used to see more time-series information from the dataset. Also, data collected from other sensors such as gyroscope and magnetometer can be used to see if that will enhance the overall performance and will help in recognizing the activity of the person more accurately.

CHAPTER IV - VIDEO-BASED ACTIVITY RECOGNITION IN SPORTS

4.1 Motivation

Body-worn sensors-based activity recognition has many advantages and has shown great improvement in real-world fields like smart homes, elderly care, and healthcare. But in some fields, such as sports the use of these sensors to collect real-time data is not feasible and practical. Although several studies have been performed with these sensors, players feel uncomfortable when they must wear these several devices on different parts of their body, not only in-game but also in training sessions also. The acceptance of this to measure the performance of the players and recognize the activity of the players without any extra burden to the players has been one of the challenges of sensor-based activity recognition. To address this issue the use of video-based activity recognition has shown a great interest for sports analysis. With easily availability and use of video capturing devices such as high-definition cameras, a smartphone with a high-resolution camera, and the increasing use of CCTV cameras for surveillance, the availability of video data has been increased rapidly. With commercial demand of sports broadcast due to the rapid growth of video transmission and global market, the use of Artificial intelligence and machine learning has played sports analysis large, diversified, and rapidly growing field for broadcasting applications. Not only for broadcasting, to boost the performance of the players, but the performance analyst also needs to go through the recording of the players for long hours to identify the activities of the person such as player movement, time of the specific activities. Manually identifying such activity from the video need lot of effort and time. The automatic system which can identify the player movements and activities from the recording that provides vital information to the coach and management staff to enhance the performance of the player by implementing in training can be helpful to improve the team. The objective of this research is to develop an automatic activity recognition system that is capable to automatically identify the activity of the people to classify different sports using the easily available video recording of the sports. As some of the factors that affect the vision-based approach are image quality, external lighting environment, illumination changes, and image resolutions we select those datasets that have been recorded and broadcasted in different broadcasting media.

4.2 Dataset Used

In this section, we present three sports activity datasets to classify sports activity.

4.2.1 Olympic sports dataset

The Olympic sports dataset[108] contains a total of 783 videos of different activities performed in 16 sports. The video sequences were obtained from YouTube and the annotation of the class labels was done with the help of Amazon Mechanical Turk. The activities related with sports are class labels into 16 different labels: high jump(67), long jump(46), triple jump(21), pole vault(40), discuss throw(63), hammer throw(46), javelin throw(25), shot put(63), basketball layup(50), bowling(49), tennis serve(39), platform diving(57), springboard diving(46), snatch weightlifting(49), clean and jerk(66), and gymnastic vault(56).

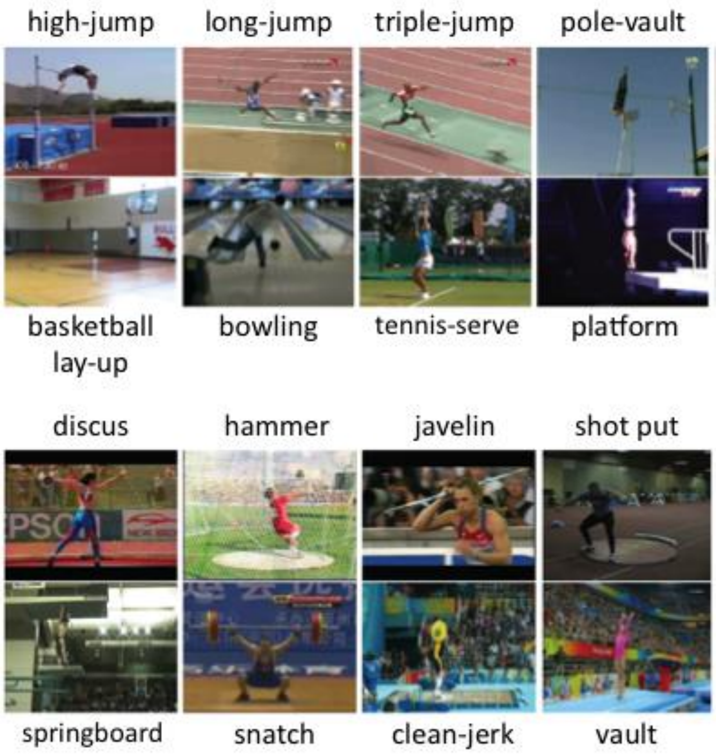


Figure 4.1 *Sample frame from 16 activities of Olympic sports dataset.*

4.2.2 Sports video in the Wild (SVW)

This dataset [117]consists of 4200 videos captured from ordinary users of the coach's eye smartphone app for sports training developed by TechSmith Corporation. This dataset has 30 different categories and 44 action categories where each video is annotated with the sports genre. The average number of videos per category is around 110 videos.

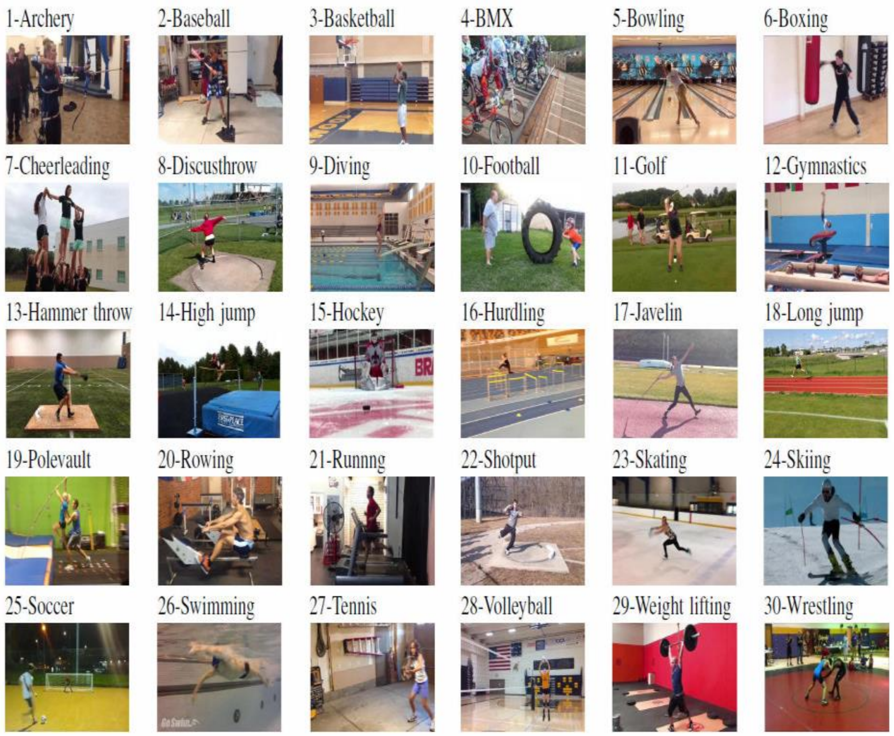


Figure 4.2 *Sample frames of all 30 activities of SVW sports dataset.*

4.2.3 UCF 101 sports categories

This UCF 101 action recognition dataset[95] consists of action videos of 101 different actions divided into five types: human-object interaction, body motion only, human interaction, playing a musical instrument, and sports. As there is a lack of a large dataset consisting of only sports-related activities and as we are interested in the study of the classification of different sports-related action, we have only considered 50 different action groups in sports categories. As horse riding and horse racing have similar activities it is placed on a single class so the different classes for this dataset in our experiment is 49. This has provided us with a large dataset with a higher number of class labels than our previous two datasets. This video in this dataset is also collected from YouTube.

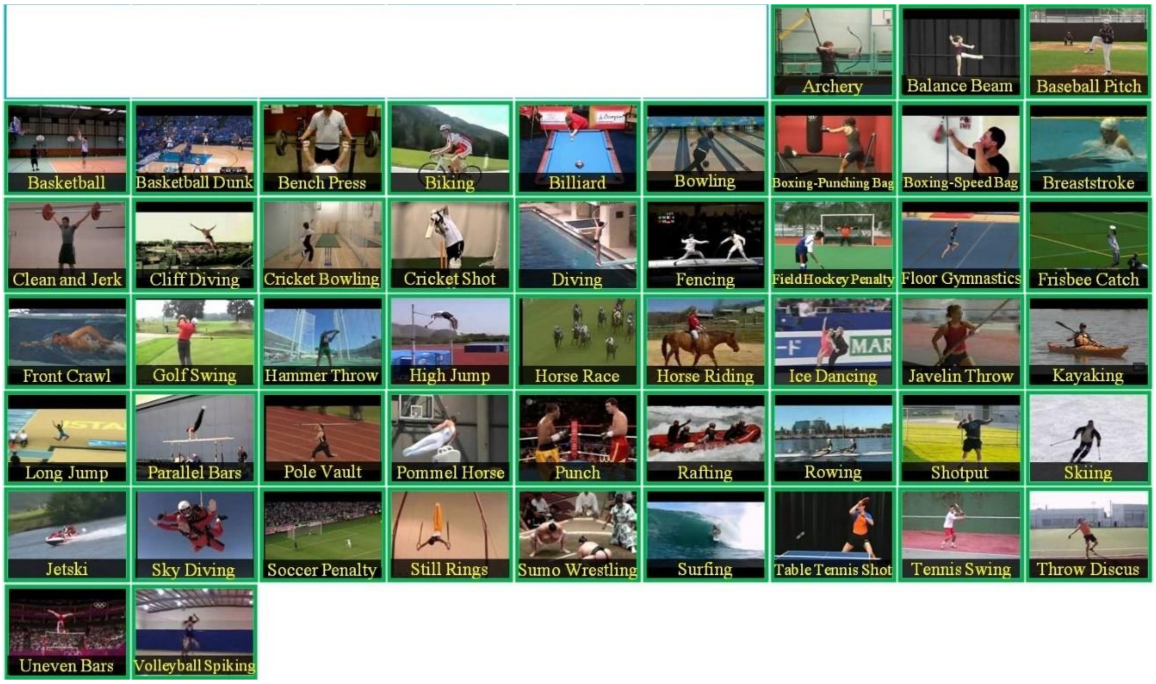


Figure 4.3 *Sample frame of all the activity in sports categories in UCF 101 dataset.*

Table 4.1 *Comparison of these three datasets.*

	Olympic Sports dataset	SVW	UCF101 sports categories
# of videos	783	4200	6671
# of class	16	30	49
source	YouTube	Smartphone & tablets	YouTube
Camera vibration	NO	Yes	No

4.3 Deep learning architecture used.

For our experiments, we used two state-of-the-art approaches 3D CNN networks: C3D[118] and C3D with LSTM. For the experimental purpose, we have used the C3D model pretrained[119] with Sports 1M dataset[90]. Sports 1M dataset consists of 1,133,158 YouTube videos annotated with 487 sports labels.

4.3.1 C3D architecture

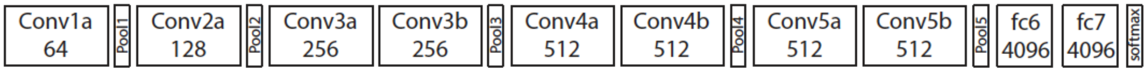


Figure 4.4 *C3D Architecture.*[119]

C3D architecture is a 3D deep convolutional neural network that consists of a 3D convolutional layer of kernel size 3×3×3 followed by a pooling layer of 2×2×2. The feature of this architecture is it extracts both spatial and temporal components of the motion of objects, humans, and scenes of the input video. The C3D architecture consists of 8 convolutional layers, 5 pooling layers, and 2 fully connected layers as shown in figure 4.4. The fully connected layer has the size of 4096 dimensions with the softmax layer for classification. To implement C3D architecture for pretrained with our dataset, we remove the last softmax layer and add a new softmax output layer with a number of classes of the respective dataset.

4.3.2 C3D + LSTM architecture

In this architecture, the hybrid model of C3D along with the LSTM architecture is proposed. For LSTM we use one bidirectional LSTM network along with one dense layer followed by a SoftMax layer. The general architecture of this hybrid model is shown in figure 4.5.

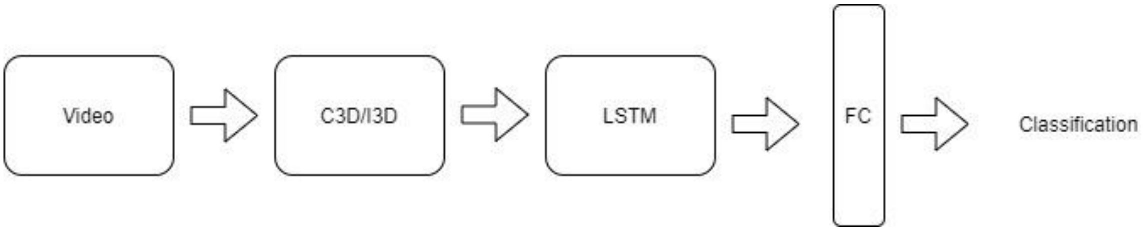


Figure 4.5 *Proposed hybrid approach*

4.4 Methodology

For the experimental purpose, we have implemented two approaches. One is fine-tuning the pretrained C3D model and extracting features from these models and use those features with a bidirectional LSTM network for classification. In this section, we will describe the application of transfer learning for video classification and these two approaches in detail.

In most of the previous research, transfer learning has been extensively used with image data. Most of the architecture that has been trained on ImageNet[120] has shown great development in image classification[51][54][121]. In video classification for action recognition, transfer learning has been implemented for individual scenes[119][69]. The purpose of transfer learning is to transfer the knowledge gain from training the model with a large dataset of similar nature and use it in a particular new dataset. Hence features learned from the source dataset are being used to test in a target dataset [122]. To build a network with huge video data, we need an enormous amount of memory space and computing power. Hence, we have used the method of Transfer learning in which we have used model that has been pre-trained on huge dataset such as Sports 1M dataset and kinetics dataset to extract high-level features.

For this experiment, we have used the C3D model pretrained on Sports 1M dataset[90] [119] and used the transfer learning method to fine-tune our three sports-related datasets. Figure 4.6 represents the complete Keras model representation of the C3D architecture which contains 5 3D convolutional layers, followed by a three-dimensional max-pooling layer and 3 fully connected layers. The last layer is the output layer with softmax classification of output class labels of 487 categories as Sports 1M dataset has

1million videos from YouTube that have been classified into 487 different categories. The second figure in Figure 4.6 represents the model architecture of our model where it consists of all the layers from the C3D architecture except the last fully connected SoftMax output layer. The last fully connected layer is replaced with a new fully connected layer for the activity classification based on the number of classes of each dataset. The figure shows the output SoftMax layer with the number of classes 16. To make our input more consistent with the input used to train the C3D model, we generated 16 frames from each video clip with a dimension of 114×114 with channel 3.

A second approach is a hybrid approach using C3D pre-trained model for low-level spatial-temporal extraction and then use LSTM for high-level temporal features extraction. For this, the features learned from the pretrained C3D model are given as an input to the bidirectional LSTM network. While training only the LSTM network is trained, freezing the pertained layers of the C3D model.

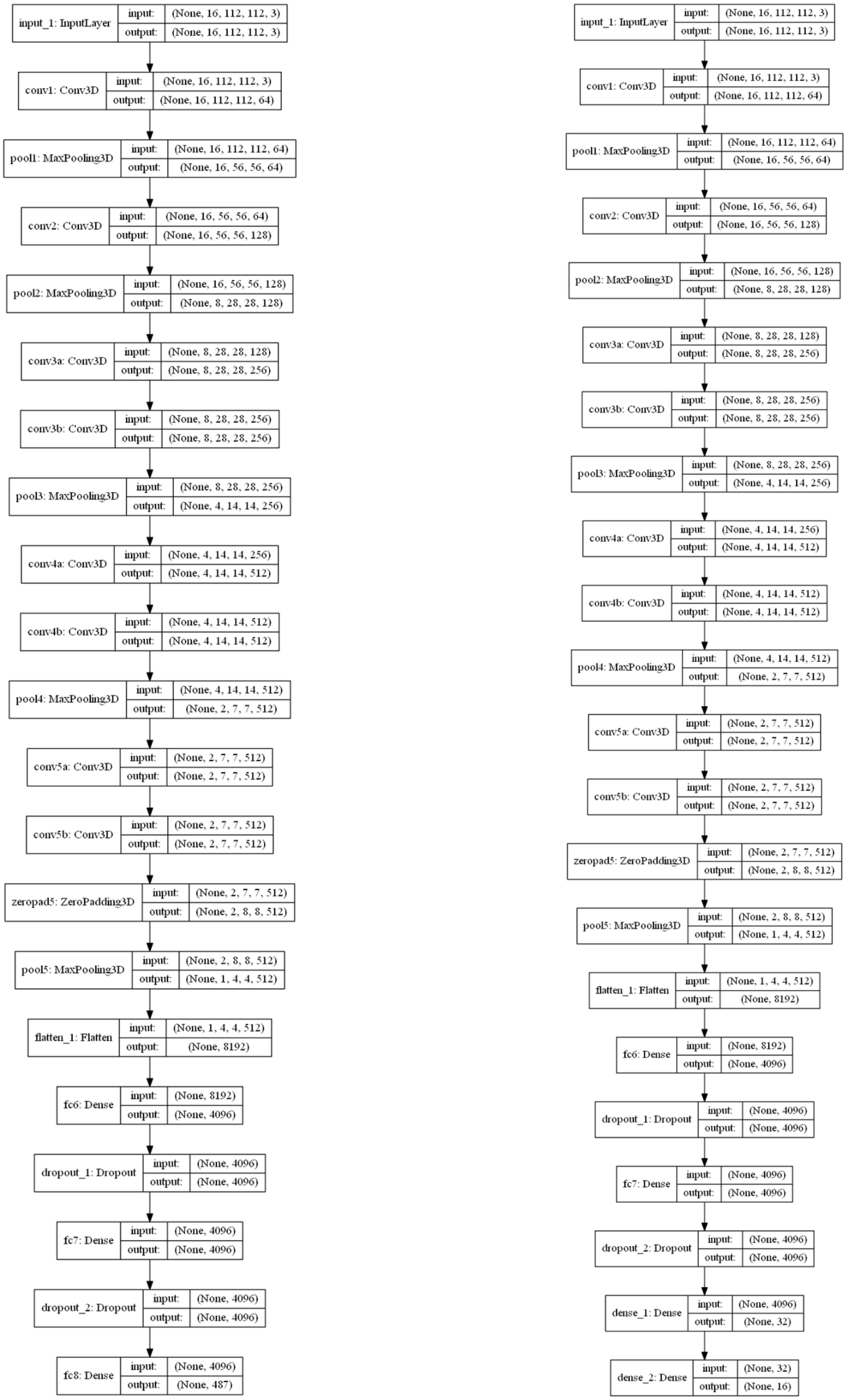

input_1: InputLayer
input: (None, 16, 112, 112, 3)
output: (None, 16, 112, 112, 3)
conv1: Conv3D
input: (None, 16, 112, 112, 3)
output: (None, 16, 112, 112, 64)
pool1: MaxPooling3D
input: (None, 16, 112, 112, 64)
output: (None, 16, 56, 56, 64)
conv2: Conv3D
input: (None, 16, 56, 56, 64)
output: (None, 16, 56, 56, 128)
pool2: MaxPooling3D
input: (None, 16, 56, 56, 128)
output: (None, 8, 28, 28, 128)
conv3a: Conv3D
input: (None, 8, 28, 28, 128)
output: (None, 8, 28, 28, 256)
conv3b: Conv3D
input: (None, 8, 28, 28, 256)
output: (None, 8, 28, 28, 256)
pool3: MaxPooling3D
input: (None, 8, 28, 28, 256)
output: (None, 4, 14, 14, 256)
conv4a: Conv3D
input: (None, 4, 14, 14, 256)
output: (None, 4, 14, 14, 512)
conv4b: Conv3D
input: (None, 4, 14, 14, 512)
output: (None, 4, 14, 14, 512)
pool4: MaxPooling3D
input: (None, 4, 14, 14, 512)
output: (None, 2, 7, 7, 512)
conv5a: Conv3D
input: (None, 2, 7, 7, 512)
output: (None, 2, 7, 7, 512)
conv5b: Conv3D
input: (None, 2, 7, 7, 512)
output: (None, 2, 7, 7, 512)
zeropad5: ZeroPadding3D
input: (None, 2, 7, 7, 512)
output: (None, 2, 8, 8, 512)
pool5: MaxPooling3D
input: (None, 2, 8, 8, 512)
output: (None, 1, 4, 4, 512)
flatten_1: Flatten
input: (None, 1, 4, 4, 512)
output: (None, 8192)
fc6: Dense
input: (None, 8192)
output: (None, 4096)
dropout_1: Dropout
input: (None, 4096)
output: (None, 4096)
fc7: Dense
input: (None, 4096)
output: (None, 4096)
dropout_2: Dropout
input: (None, 4096)
output: (None, 4096)
fc8: Dense
input: (None, 4096)
output: (None, 487)
input_1: InputLayer
input: (None, 16, 112, 112, 3)
output: (None, 16, 112, 112, 3)
conv1: Conv3D
input: (None, 16, 112, 112, 3)
output: (None, 16, 112, 112, 64)
pool1: MaxPooling3D
input: (None, 16, 112, 112, 64)
output: (None, 16, 56, 56, 64)
conv2: Conv3D
input: (None, 16, 56, 56, 64)
output: (None, 16, 56, 56, 128)
pool2: MaxPooling3D
input: (None, 16, 56, 56, 128)
output: (None, 8, 28, 28, 128)
conv3a: Conv3D
input: (None, 8, 28, 28, 128)
output: (None, 8, 28, 28, 256)
conv3b: Conv3D
input: (None, 8, 28, 28, 256)
output: (None, 8, 28, 28, 256)
pool3: MaxPooling3D
input: (None, 8, 28, 28, 256)
output: (None, 4, 14, 14, 256)
conv4a: Conv3D
input: (None, 4, 14, 14, 256)
output: (None, 4, 14, 14, 512)
conv4b: Conv3D
input: (None, 4, 14, 14, 512)
output: (None, 4, 14, 14, 512)
pool4: MaxPooling3D
input: (None, 4, 14, 14, 512)
output: (None, 2, 7, 7, 512)
conv5a: Conv3D
input: (None, 2, 7, 7, 512)
output: (None, 2, 7, 7, 512)
conv5b: Conv3D
input: (None, 2, 7, 7, 512)
output: (None, 2, 7, 7, 512)
zeropad5: ZeroPadding3D
input: (None, 2, 7, 7, 512)
output: (None, 2, 8, 8, 512)
pool5: MaxPooling3D
input: (None, 2, 8, 8, 512)
output: (None, 1, 4, 4, 512)
flatten_1: Flatten
input: (None, 1, 4, 4, 512)
output: (None, 8192)
fc6: Dense
input: (None, 8192)
output: (None, 4096)
dropout_1: Dropout
input: (None, 4096)
output: (None, 4096)
fc7: Dense
input: (None, 4096)
output: (None, 4096)
dropout_2: Dropout
input: (None, 4096)
output: (None, 4096)
dense_1: Dense
input: (None, 4096)
output: (None, 32)
dense_2: Dense
input: (None, 32)
output: (None, 16)

Figure 4.6 *Model architecture for C3D (left) and new model structure(right)*

4.5 Experimental Results

For experimental analysis, we perform the experiments on C3D architecture to fine-tune a pretrained model with the training set on our three sports-related datasets. For training, we used the nAdam optimizer with its default parameters with a learning rate of 0.00001. The loss function chosen for this is categorical cross-entropy. We perform the experiments for 300 epochs with early stopping patience of 20. But with each epoch, the model was trained with training data, and then validate with the validation dataset. The performance of the model is evaluated with the validation loss in each epoch. The model is saved with each improved performance and at the end of the epoch, the model with the best result is used in testing with testing data. The best model with the minimum loss function at the validation data is saved. Although the model is trained for a maximum of 300 epochs the model which shows the best result throughout the epoch has been used for testing. The details of the parameter used for the architecture are shown in table 4.2.

Besides that, we also perform our experiments with 5-fold stratified cross-validation for the pretrained fine-tuning approach. For that, all the dataset was divided into five folds based on the contained class. In each iteration, one-fold was used for testing, and among the remaining 4-fold, 25% of data was used for validation while others were used for training our model. After each iteration, the test set change and will repeat the same process. We did this with fine tuning pretrained approach. Table 4.3 shows the overall accuracy for three datasets using the C3D pretrained model with the fine-tuning approach.

Table 4.2 *Model parameter for C3D architecture.*

	C3D Architecture
# of frames from each video	16
Frame dimension	112×112
Learning rate	0.0001
Optimizer	Nadam
Epochs	100
Early stopping	Yes
Patience	20
K fold	5
Pretrained	Yes (Sports 1M dataset)

Table 4.3 *Result of 5-fold cross-validation for three datasets using C3D pretrained fine-tuning method.*

	Olympic		UCF100		SVW	
	Top1	Top3	Top1	Top3	Top1	top3
Fold 1	61.78%	85.35%	68.76%	87.82%	49.16%	71.02%
Fold 2	55.41%	82.16%	68.22%	85.31%	50.77%	72.88%
Fold 3	54.77%	82.80%	67.39%	84.26%	48.87%	73.62%
Fold 4	58.33%	84.61%	63.19%	83.81%	47.68%	70.90%
fold5	63.46%	85.89%	69.11%	88.91%	51.72%	71.81%
Average	58.75%	84.17%	67.34%	86%	49.64%	72.04%

For the second approach, we extracted features from the C3D model and then passed it to the bidirectional LSTM model for temporal feature extraction. The weights from the c3D model are freeze during training the hybrid model. Figure 4.7 shows the accuracy and loss curve for training and validation data along with the normalized confusion matrix plot for the C3D+LSTM network for the Olympic Sports dataset. Similarly, Figure 4.8 and Figure 4.9 represent the accuracy and loss curve for training and validation data with confusion matrix for dataset SVW and UCF100 sports dataset.

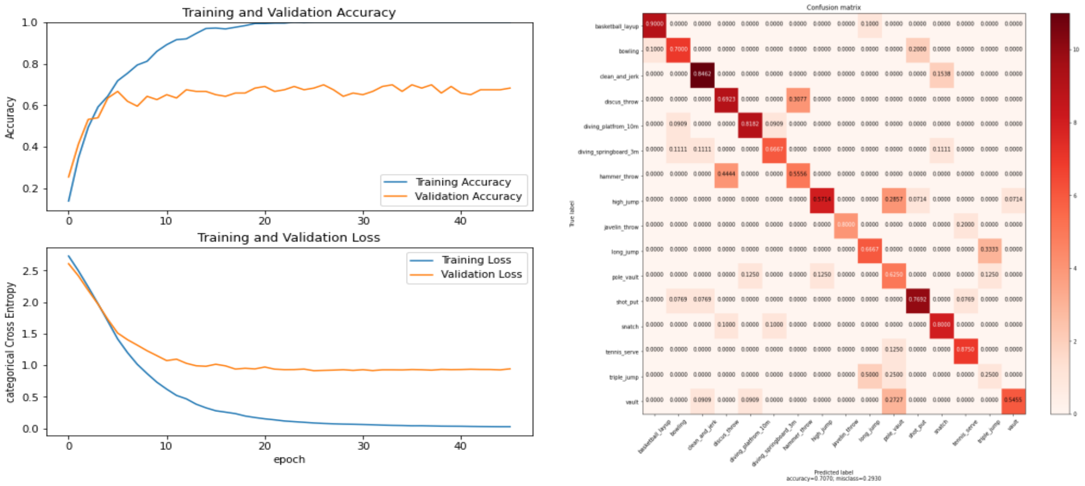


Figure 4.7 *Accuracy and loss curve for training and validation data(left), a confusion matrix plot(right) for Olympic dataset with pretrained C3D+LSTM model*

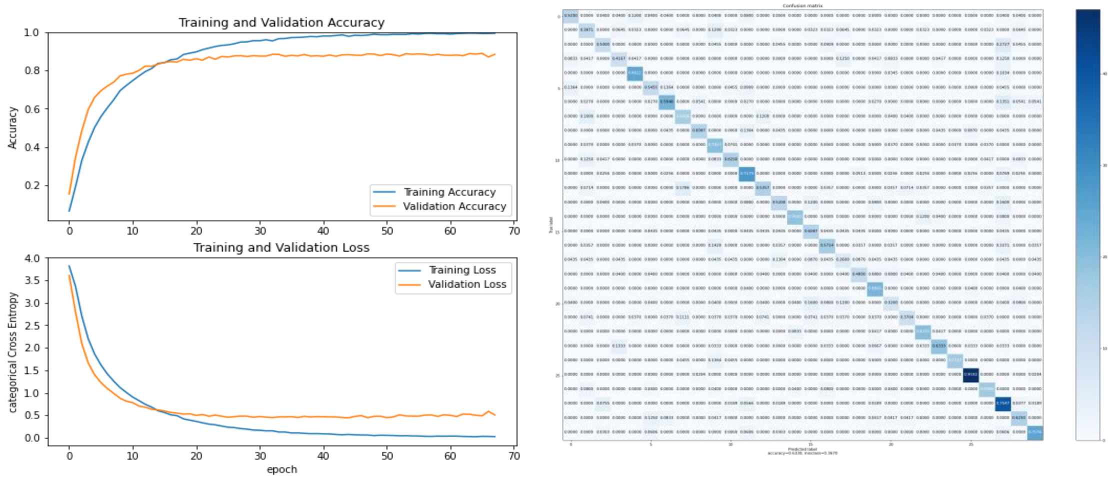


Figure 4.8 *Accuracy and loss curve for training and validation data(left), confusion matrix plot(right) SVW sports dataset with pretrained C3D+LSTM model.*

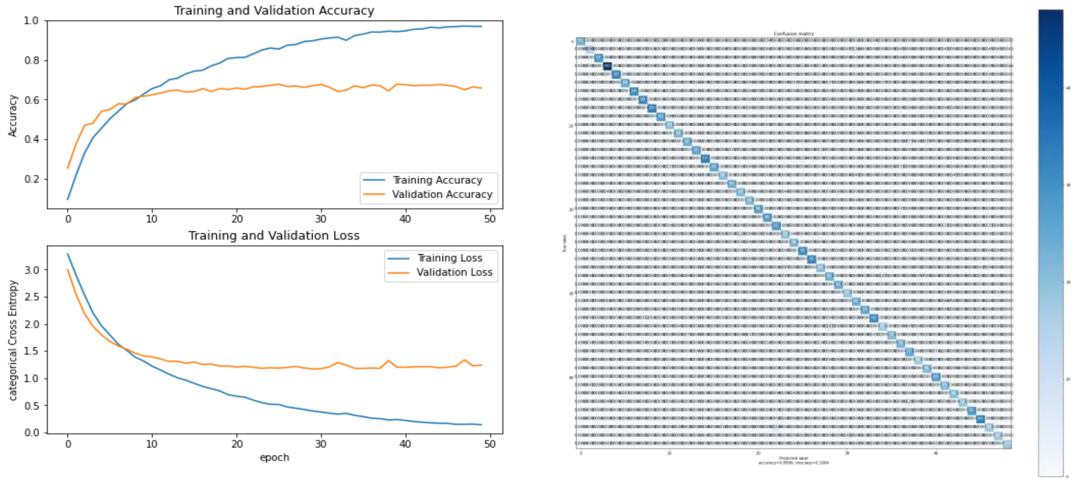


Figure 4.9 *Accuracy and loss curve for training and validation data(left), confusion matrix plot(right) for UCF100 sports dataset with pretrained C3D+LSTM model.*

The overall accuracy for all the datasets for C3D fine-tuning and hybrid model for all the datasets is shown in Table 4.4. here we can see that UCF100 sports dataset have higher accuracy with compared to other datasets for both the approach. The hybrid approach outperforms the fine-tuning pretrained C3D approach in all the datasets.

Table 4.4 *Accuracy table for all the models used for all three datasets.*

	Olympic dataset		UCF 100 sports		SVW dataset	
	Top1	Top3	Top1	Top3	Top1	Top3
C3D+fine tuning	58.75%	84.17%	67.34%	86%	49.64%	72.04%
C3D+LSTM	70.70%	92.36%	89.96%	96.40%	63.30%	82.42%

As the C3D model was pretrained with a sports 1M dataset that has videos from YouTube, the UCF100 sports dataset like this has shown higher results than the other two datasets. The size of the Olympic dataset is very low as compared to others which can be the reason for lower performance compared to others. SVM dataset has the lowest accuracy than the other two in both the approach. The dynamic nature and vibration in the camera can have added more challenges in recognizing the activities for this method.

4.6 Discussion and Conclusions

Hence in this section, we implement a transfer learning approach for activity recognition where a model that has been pretrained to some bigger dataset and use that trained model to a smaller dataset of the same nature. In our experiments, we used the C3D model pretrained on a bigger dataset related to sports 1M and used the information gained on this for a small sports dataset as compared to the pretrained dataset. This method is very helpful to reduce high computational requirements and energy and memory consumption to train a model. Also, using this method can reduce the computational power requirements and be used for interring field knowledge transmission.

In our experiment, we applied this technique for our three sports-related datasets and used two approaches for activity classification. The output shows comparable results in all the datasets for the Top 3 performance analysis. The hybrid model has outperformed the fine-tuning C3D model for all three datasets. More deep knowledge transfer and other models trained on other bigger datasets such as I3D model pretrained on kinetics and ImageNet dataset can also be tested for better performance.

CHAPTER V – ACTIVITY RECOGNITION IN BASKETBALL

5.1 Motivation

Among different fields where the application of machine learning and deep learning has proved to be very successful and useful, sports have also been one of the most popular areas where the application of Artificial Intelligence (AI) has shown great potential. For monitoring player fitness, to detect the injury that can occur to players during training or while playing, analyzing the game in real-time, making strategy during play, and analyzing player performance AI or AI enables devices to have been used. Not only on-field or players, AI and computer vision has been popular in sports marketing, sports TV broadcasting, sports coverage, and broadcasting. In most cases, players will wear AI-enabled wearable devices with sensors, which will transmit data with players' movement, position and the data received from the sensors is being used for analysis and activity recognition. But in real-world game, the use of sensors or body-worn devices make it hard for players in their movement or concentrate during the game. This can be resolved if we can use videos in place of sensor data for the activity recognition that occurs in the game This can help to develop an automated system that can help to gather the information that can be helpful to analyze players, provide on-time analysis, and aid referee to make decisions during the game. Although recent research has been conducted with special multiple camera systems [123] that try to make the activity recognition more accurate and efficient, the operational complexity and cost have made it impossible to implement it in all sports. Hence, the main objective of this research has been to prepare a basketball dataset from the video that has been broadcasted live from the broadcasting media which is easily available to classify scoring activities during the game. This final objective will be to develop an automated system that can recognize the scoring activities such as two-point shots, three-point shots are free throw, and other activities such as dribbling, passing, etc.

5.2 Literature review

This section explains some of the existing sports-related datasets and methods that have been used for activity recognition related to this activity. Many activity recognition datasets include daily activities such as walking, running, climbing stairs, sitting, etc. Among some other datasets such as UCF101[95], kinetics[91], HMDB51[89], they include activities related to sports such as playing basketball, horse riding, throwing, kicking, playing cricket, etc. This dataset mainly consists of videos or images of players performing different activities related to their specific sports and has been included as action recognition in these datasets. Some datasets such as Ilur news text[124], YT-UGC[125], and AG news[126] where the dataset is divided into different categories, and among them one of the categories has been sports. Some sports-specific dataset has also been developed which consists of images or videos clips. UCF sports[86] consists of video clips collected from sports broadcast networks. Olympic sports[108], sports-1m[90], SVW[117] and wang et al[127] etc., that are used for sports classification. These datasets contain a large collection of video clips from a different source of variable length which includes multiple sports actions. Also, some dataset contains information about a specific sport of different domain such as basketball[123], table tennis, golf [128], soccer[129] and hockey[130]. In[130] they have prepared their dataset for classifying four different activities related to hockey: free hit, goal, penalty corner, and long corner. They used pretrained deep learning model to classify these activities with image data. They consist of the information about some specific actions related to those sports that are used for classifying the recognizing different activities performed by the players. For example, a dataset like NBA sportsUV contains the player and ball trajectory from 631 games from the 2015-2016 NBA season. Also, videos of the players during the games is used for analyzing the player movement by player tracking, behavior of player during game and has been used for recognizing these action of a basketball players [131]. In UIT-VILC[132] sports related to sports played with a ball are included which has been studied for image captioning. By analyzing the score information and the modeling excitement, to generate highlights from the video is discussed for basketball games in this paper[133]. In this method first they classify the video as play and non-play shots and with score automatically extracted from the video games will select whether to include that on the highlight video or not. In [134] analysis in the basketball trajectories was analyze to predict whether the three point shot in the game is successful or not. They used recurrent neural networks to learn the trajectory of a basketball with full set of features like angle and velocity to predict the three-point shot. Also, to recognize the offensive NBA plays, machine learning with deep neural network and RNN is used for classification using the tracking data of the player during the game. It also showed good recognition rates while training with the data and testing the model with the data from another seasons [135].

For the classification of these activities with this dataset, machine learning and deep learning architecture have been used. As the change in action has information in a long range of video frames, temporal information is needed to recognize the activities. For this RNN network such as LSTM has been used in the different study[136][137][138]attention. Also, optical flow information has been used to provide temporal information in one stream and the RGB data for spatial information in the other stream, and a two-stream network[61][139][60] has been used for classification in recent times.

5.3 Dataset Used

Many video-based human activity recognition datasets[43] consist of activities based on daily activities such as walking, running, climbing stairs, jogging, etc. In some datasets [89][95][91][109], some sports-related activities performed by the subject such as playing basketball, throwing a baseball, juggling, or playing soccer, etc. have been classified as different activities performed by the person. Some datasets have been developed using sensors where the statistical analysis of data captured from the sensors has been used for analyzing player performance. Also, many sports-related visual datasets [86][108][90] have been developed to classify different sports based on the video captured during the game. Some of the sports that have been included in these datasets are basketball, baseball, tennis, badminton, football, horse riding, etc. This dataset includes videos or images that have been collected from YouTube, broadcasting media, smartphones, or other recording devices. This dataset has provided great contributions in the field of computer vision and activity classification to classify different sports. But there is a lack of a dataset that contributes towards the classification or recognition of different activities related with a single game. For example, if we can develop an automation system that are able to recognize the activity that has been performed on the game and will be able to automatically update the score, replace the official or at least assist them to make correct decision can greatly improve the sports activities. For that, we prepare our own dataset that includes the different scoring activities performed in a basketball game. Also, we have prepared from our dataset from the videos that has been broadcasted from the media during the game.

5.3.1 Data Collection

The following section defines the data collection process and preprocessing techniques that have been used to prepare the basketball dataset. The steps involved in the data collection are:

Step1: A collection of videos broadcasted on broadcasting media.

The first step was to collect the videos of the live NBA games that have been broadcasted on different broadcasting channels. These full videos of games consist of all the coverage of the game along with advertisements, flashing graphics, replays, highlights, interviews, scores, and other displays. The whole game video is like what we have seen during the live games broadcasted on our TV. The quality of these videos recorded at first is in HD.

Step 2: Manually store the label and timestamp.

The second step is to check each video and manually store the timestamp in the video of the point when the ball is near the rim of the basketball court. That is when a player makes a shot the time at a point when the ball is just above the rim is being stored. The timestamp consists of Hours (HH), minutes (MM), and second (SS) of that point of time. Then along with the time, the range of the shot, whether the point of throw was from the three-point range, two-point range, or a free throw range, and the scoring status whether it is a score or a miss. Hence the list consists of the timestamp and manual labels of that activity at that timestamp.

Step 3: Generate clips.

Once we have the timestamp of that activity and its labels, then a video clip of a length of 6 seconds is generated. The clips will have the content 3 seconds before and 3 seconds after the timestamp. The reason to choose the length of the clips is such that the clips include all the actions related to the scoring activity in them. Each clip has information about the range from which the shot was taken, whether it has the rebound or multiple attempts to score, whether there was a score or not. Figure 5.2 shows the frame from the clips at the point the shot is being taken, the point of time when the ball is near the rim of the basket, and the time when the ball has either score or not score. The average number of clips generated from each full basketball game video is around 200. We generate the clips from 51 such NBA games.

Step 4: Data reduction and labeling.

Once we generate the clips from the basketball videos, we reduce the clips from HD to a dimension of 320×240 pixels value. The reason behind this is to reduce the size of the dataset without losing much information from the clips such that it requires less memory requirement in the experimental process. Then the clips are manually labeled into 8 different classification labels. These labels are Two-point make(2p1), Two-point miss(2p0), Three-point makes (3p1), three-point miss(3p0), Mid-range make (Mp1), mid-range miss (Mp0), Free throw makes (FT1), and Free throw miss (FT0). The clips are nomenclature in such a way that it contains information of the video number, timestamp and the labeled of the clips.

Step 5: Generate Optical flow clips

Once we have the video clips of different actions which are in RGB, we also generate the optical flow video clips from the RGB clips for experimental analysis of its temporal information. For this, we used the optical flow concept developed first by [140]which finds the relationship between the consecutive frames of the video clips. Using open-source library OpenCV with the Gunnar Farneback optical flow techniques [141], we generate the labeled optical flow video clips and used them in our experiments to learn the temporal flow information from the clips. Figure 5.1 shows the sample frame of different activities from the optical flow videos whereas Figure 5.2 shows the sample frames for the RGB video dataset for different activities. In figure 5.2, left to right first column represents the frame at the time of the player taking the shot, the second frame at the time when the ball reaches the rim, and the third column represents after the ball pass the rim for 8 different class labels: 2p0,2p1,3p0,3p1, ft0,ft1,mp0,mp1.(from top to bottom).

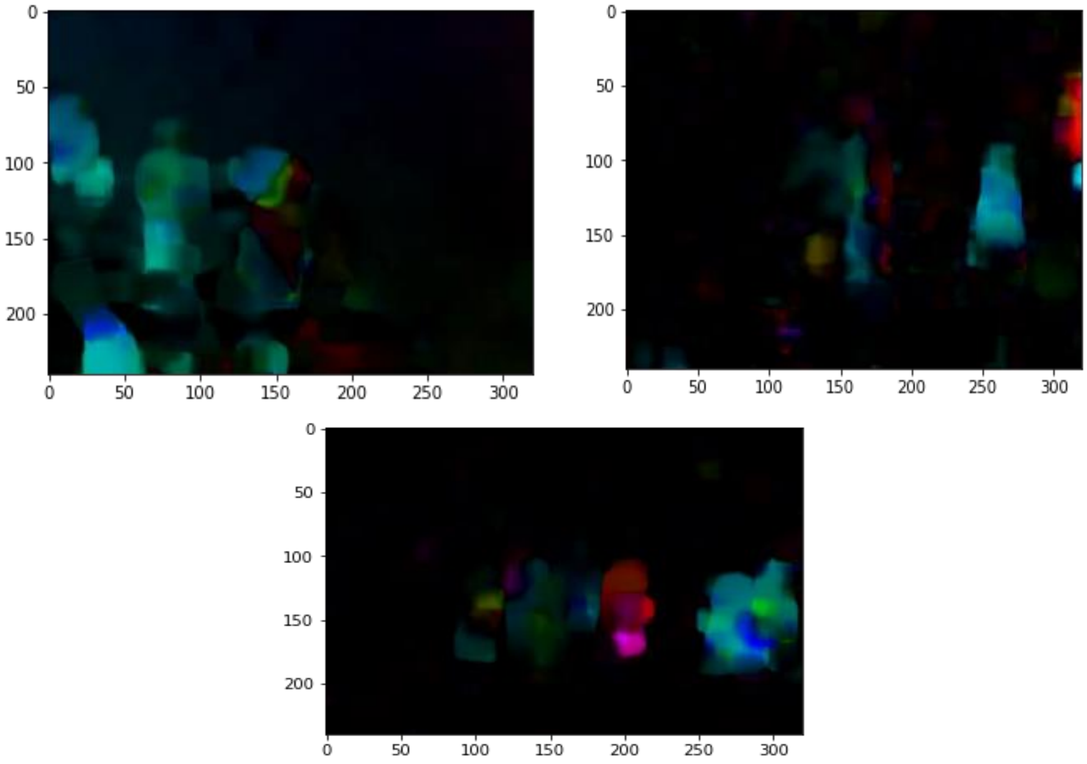


Figure 5.1 *Sample frame of optical flow video for different activity*

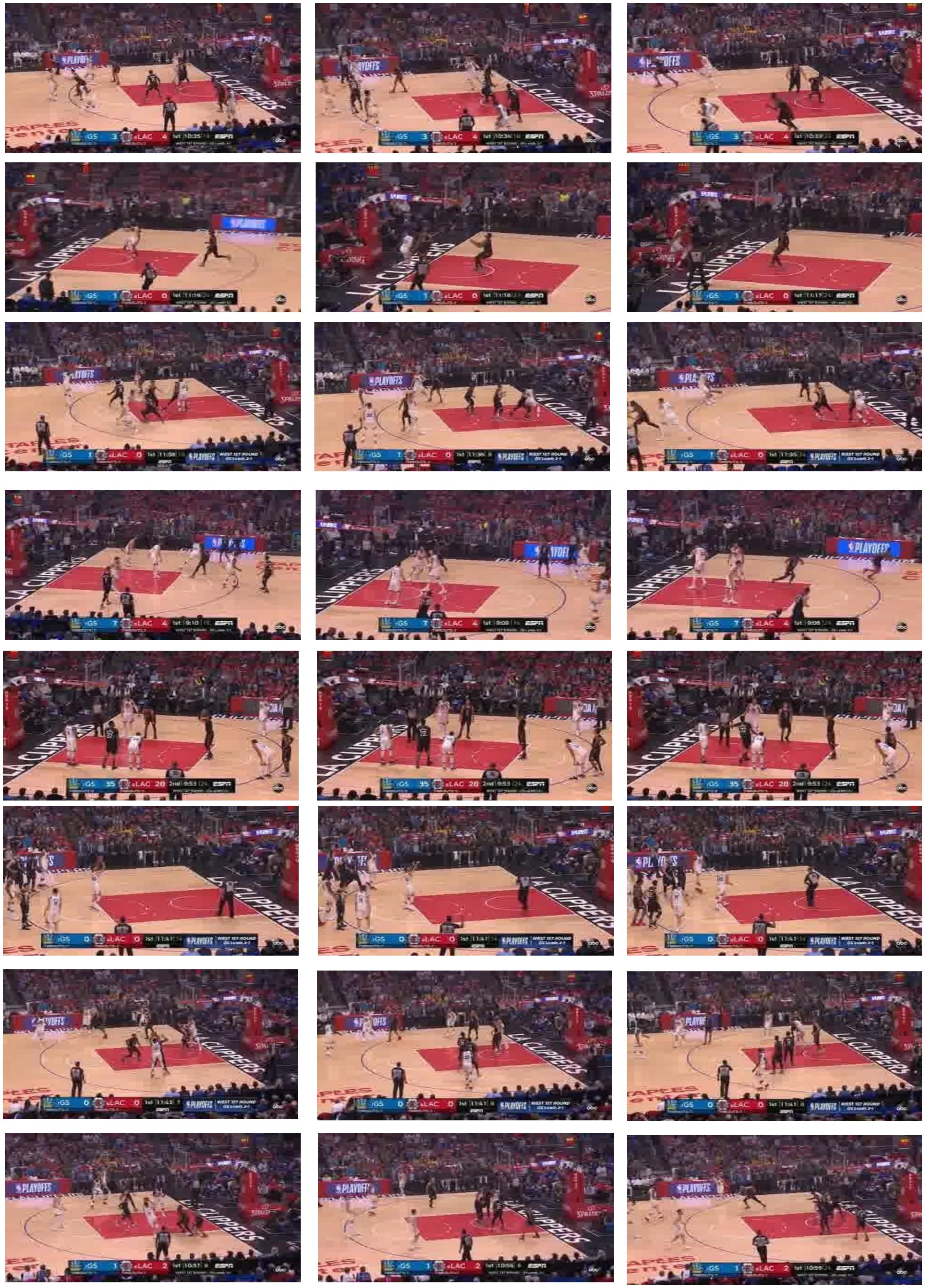


Figure 5.2 An example frames of different scoring activities of the basketball dataset.

5.3.2 Dataset characteristics

The dataset has a total of 10,311 video clips that have been generated from 51 NBA basketball games broadcasted in the broadcasting media. Thus, the video has been captured from the camera used by the broadcasting media with a third-person view. This has reduced the requirement of using multiple cameras with specific specifications in a specific position around the court for data collection and analysis.

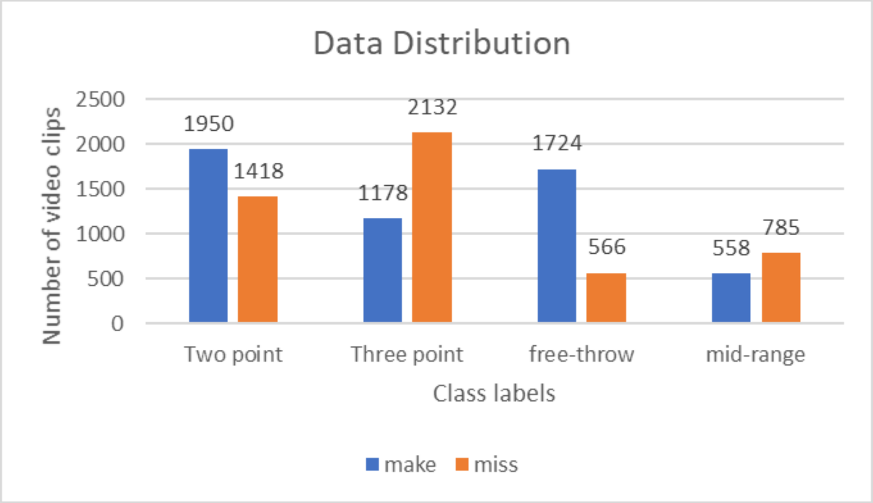


Figure 5.3 *Class distribution of the dataset based on different groups*

The whole dataset is initially labeled into 8 different class labels: two-point miss/make, three-point miss/make, Free throw miss/make, and mid-range miss/make. Depending on the study criteria., it can be categorized into different groups. Figure 5.2 represents the data distribution among different classes based on different groups. For example, to train and test the model for the range of the shot being taken, the whole dataset can be divided into four groups, Two-point, three-point, free-throw, and mid-range shot. To study the scoring of the basket if it is being scored or not, the dataset can be categorized based on made or miss. The sample frame of the RGB video dataset for each class activity is shown in figure 5.3.

The maximum number of clips among all the class is that of three-point miss (2132), which is about 20% of the entire dataset whereas the minimum number of clips is that of free-throw miss (558) and mid-range make (558) which are around 5.4% of total data. The distribution of the entire dataset is based on the player scoring activities for the 51-basketball game. This also represents the frequency of the scoring shots players plays during the game.

5.4 Deep learning architecture used.

In this research, we use two state-of-art methods that are being used for video classification. One is the 3D CNN method which is used for classifying different scoring activities for both RGB and optical flow datasets. Another one is the two-stream CNN method, which is used to analyze both the spatial and temporal nature of the dataset. The details of the used architecture and its methodology are explained in the next sections.

5.4.1 3D CNN

For our analysis for video classification, we have used 3D CNN which has 4 convolution layer blocks where each block has a convolutional layer, max-pooling layer, dropout layer, and batch normalization layer. As 3D CNN has three-dimensional convolutional kernels that can make segmentation predictions for a volumetric patch, it shows the ideal approach for video segmentation [142][143]. Figure 5.4 shows the basic structure of the 3D CNN network used in our study. From the input video, 50 frames are captured with a pixel size reduced to 80×80.

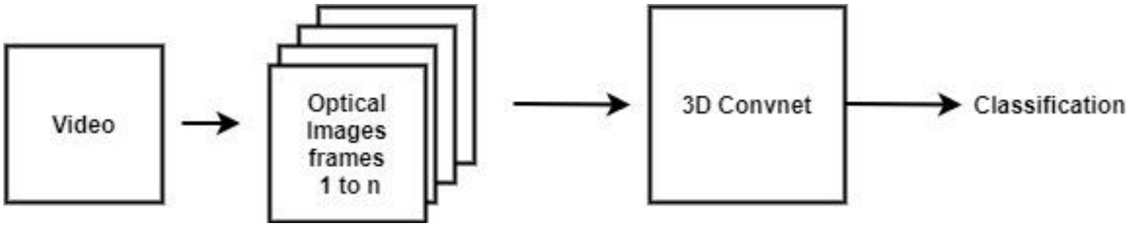


Figure 5.4 *3D CNN network*

5.4.2 Two stream 3D CNN

In two-stream networks, we train our model with RGB video frames for spatial information and another stream with optical flow video frames for temporal information. These two-network fused taking average of the predicted features before passing to the fully connected layer. We used an identical model for both the RGB video and Optical video with identical model parameters and input dimensions. Figure 5.5 shows the configuration of the two-stream network architecture used in our experiments.

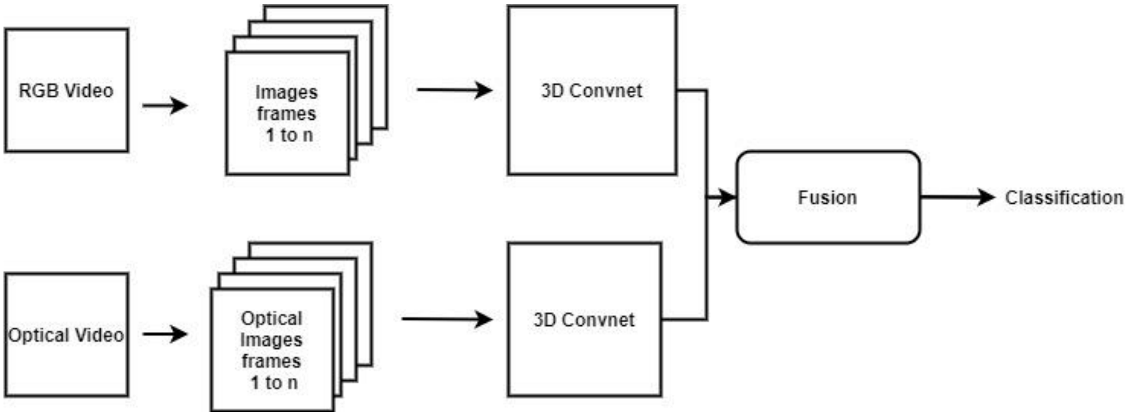


Figure 5.5 *Two stream CNN Network*

5.5 Experiments

This section explains some of our preliminary results using our dataset for baseline performance benchmark. For this, we use 3D CNN model as it is best suitable for video classification. For our experiments, we divide our dataset into 3 different groups based on the number of activity classes for classification: 2-class, 4-class, and 8-class. For each group, we again perform our analysis in RGB video input data and Optical video data. To study the effect of color and effect to use single-channel input(grayscale) or multi-channel input (RGB) we perform our experiments for two-channel inputs represented as 3D for RGB channel and 1D for grayscale in our results. Also, to study different input parameters such as background audience, court color, players jersey and movement, camera orientation, we perform our study into two classification techniques: subject dependent (SD) and subject independent (SI).

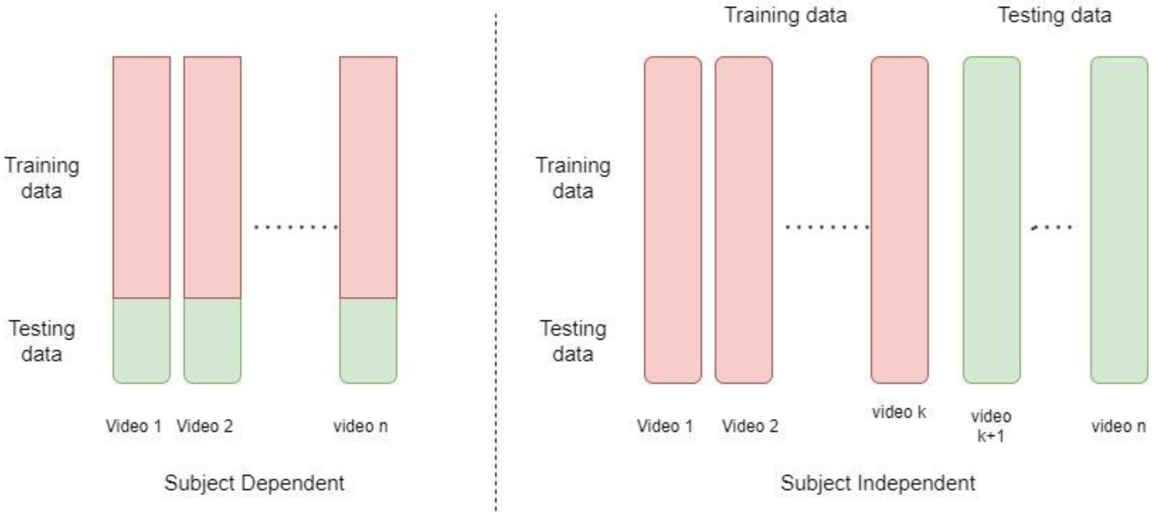
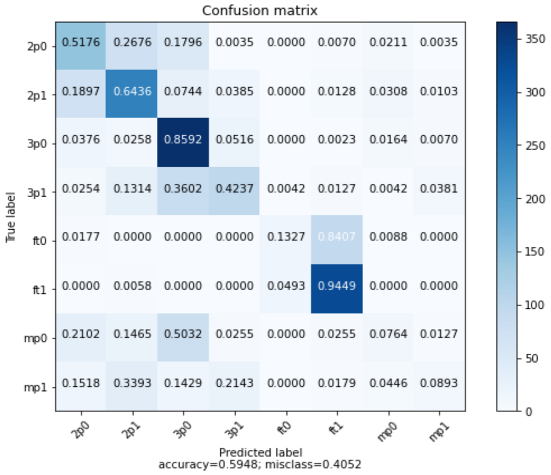


Figure 5.6 *The distribution of data to training and testing data for subject-dependent and subject-independent classification.*

To generalize our results for the different real-time scenarios, as the testing data(game) can be very much different from the training video data, subject-independent analysis can be helpful. For subject-dependent analysis, we divide all the datasets into training and testing datasets such that clips from any video may be on the training and testing dataset. Figure 5.6 represents the data distribution for training and testing for subject-dependent and subject-independent approaches. For subject-dependent analysis from n number of videos, the training and testing dataset is prepared with clips from all the videos. For subject independent analysis, clips from k number of videos were used for training the model while clips from (n-k) number of videos were used as a testing dataset. For our experiments, we used clips from 40 game videos as training data and 11 game videos as testing data. Hence the testing data is entirely different clips with different features than the training clips used to train the model. For subject dependent, the whole dataset is divided into (80/20) training and testing data and the training data is again divided into (80/20) training and validation dataset. For all the cases the normalized confusion matrix plot, confusion matrix table, and classification report are presented.

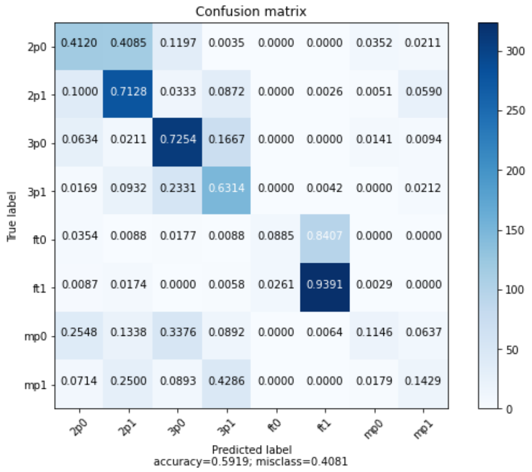
5.5.1 Subject dependent (SD)

As previously mentioned for subject-dependent classification, we divided all the datasets into training, validation, and testing datasets. We trained our model with the training dataset and validate our model with the validation dataset. The best model is saved based on the improvement in the validation loss. At the end of the iteration, the testing data is tested with the model with the best result in the validation dataset. The overall accuracy presented in the table is the accuracy of the testing dataset.



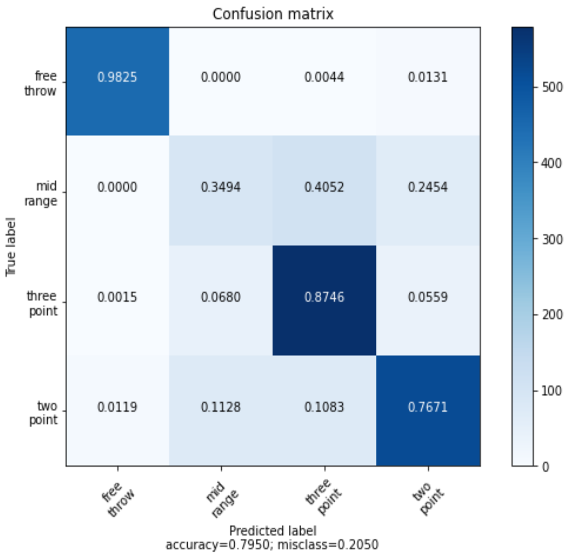
```
                    Classification Report
                    precision    recall f1-score support

     0          0.50     0.52     0.51      284
                    1            0.58     0.64     0.61
                                                  390
                    2            0.58     0.86     0.70
                                                  426
     3          0.60     0.42     0.50      236
     4          0.45     0.13     0.21      113
     5          0.74     0.94     0.83      345
     6          0.27     0.08     0.12      157
     7          0.34     0.09     0.14      112

   accuracy                          0.59     2063
macro avg          0.51     0.46     0.45    2063
weighted avg          0.56     0.59     0.55  2063
```

Figure 5.7 *SD Confusion matrix plot and classification report for class-8 RGB (grayscale) input video*



```
                    Classification Report

                precision   recall f1-score  support

     0          0.48     0.41     0.44      284
     1          0.58     0.71     0.64      390
     2          0.65     0.73     0.69      426
     3          0.47     0.63     0.54      236
     4          0.53     0.09     0.15      113
     5          0.77     0.94     0.84      345
     6          0.46     0.11     0.18      157
     7          0.25     0.14     0.18      112

   accuracy                          0.59     2063
  macro avg     0.52     0.47     0.46     2063
weighted avg    0.57     0.59     0.56     2063
```

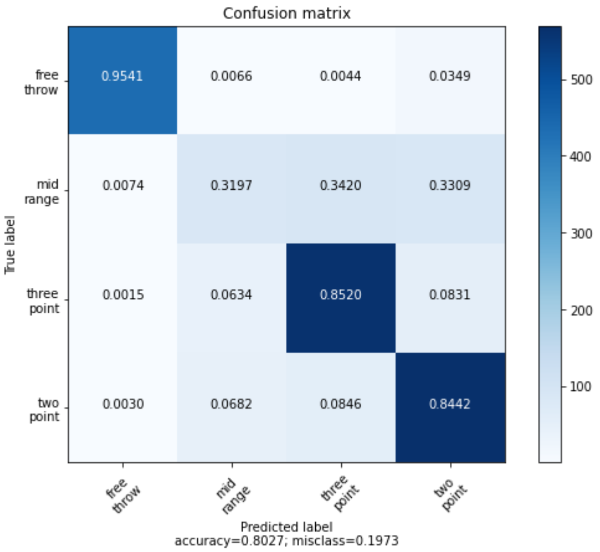
Figure 5.8 *SD Confusion Matrix plot and classification report for class-8 RGB input video*



	free throw	mid range	three point	two point
free throw	0.9825	0.0000	0.0044	0.0131
mid range	0.0000	0.3494	0.4052	0.2454
three point	0.0015	0.0680	0.8746	0.0559
two point	0.0119	0.1128	0.1083	0.7671

```
              Classification Report

            precision   recall f1-score   support

       0       0.98       0.98    0.98        458
       1       0.44       0.35    0.39        269
       2       0.76       0.87    0.81        662
       3       0.83       0.77    0.80        674

 accuracy                         0.79       2063
macro avg       0.75       0.74    0.74      2063
weighted avg    0.79       0.79    0.79      2063
```

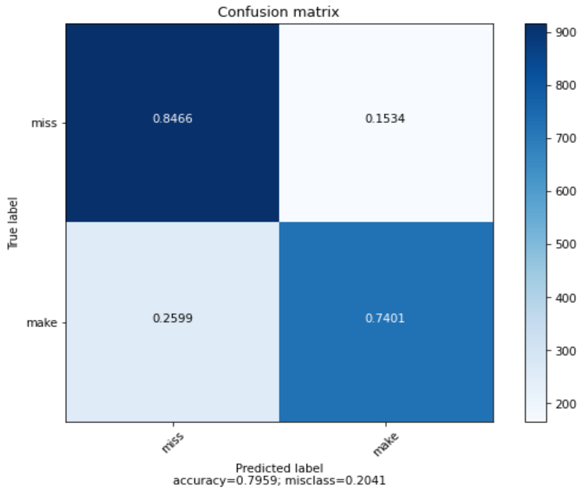
Figure 5.9 *SD Confusion matrix plot and classification Report for class-4 RGB (grayscale) input video*



	free throw	mid range	three point	two point
free throw	0.9541	0.0066	0.0044	0.0349
mid range	0.0074	0.3197	0.3420	0.3309
three point	0.0015	0.0634	0.8520	0.0831
two point	0.0030	0.0682	0.0846	0.8442

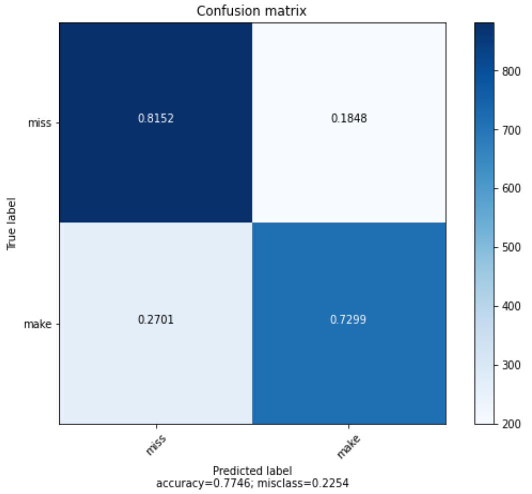
```
              Classification Report

            precision   recall f1-score   support

       0       0.99       0.95    0.97        458
       1       0.49       0.32    0.39        269
       2       0.79       0.85    0.82        662
       3       0.78       0.84    0.81        674

   accuracy                       0.80       2063
  acro avg     0.76       0.74    0.75       2063
weighted avg   0.79       0.80     0.79      2063
```

Figure 5.10 *SD Confusion Matrix plot and classification Report for class-4 RGB input video*



```
Classification Report

              precision    recall  f1-score support

           0       0.78      0.85      0.81    1082
           1       0.81      0.74      0.78     981

    accuracy                           0.80    2063
   macro avg       0.80      0.79      0.79    2063
weighted avg       0.80      0.80      0.80    2063
```

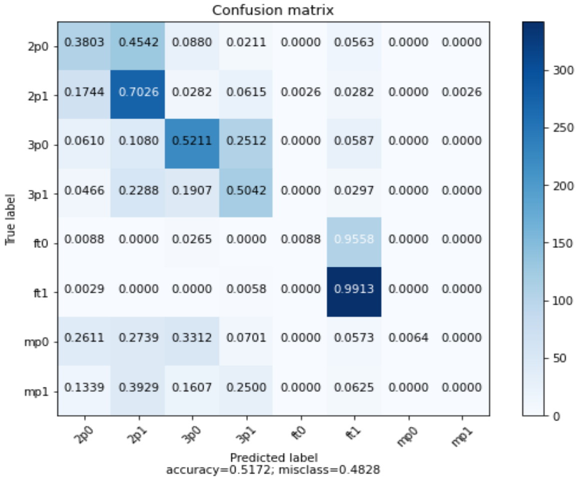
Figure 5.11 *SD Confusion matrix plot and classification Report for class-2 RGB (grayscale) input video.*



```
Classification Report

              precision    recall f1-score   support

           0       0.77      0.82      0.79    1082
           1       0.78      0.73      0.75     981

    accuracy                           0.77    2063
   macro avg       0.78      0.77      0.77    2063
weighted avg       0.78      0.77      0.77    2063
```

Figure 5.12 *SD Confusion Matrix plot and classification Report for class-2 RGB input video.*

Figure 5.7 to Figure 5.12 represents the confusion matrix and classification report for grayscale and color input video. The accuracy for 8-class classification is lower at around 59% for both 1D and 3D, whereas the accuracy is highest for class-4 and class-8 which is around 80%. The f1-score is lowest for the mid-range shot which is mostly misclassified as three-point and two-point in all the cases.

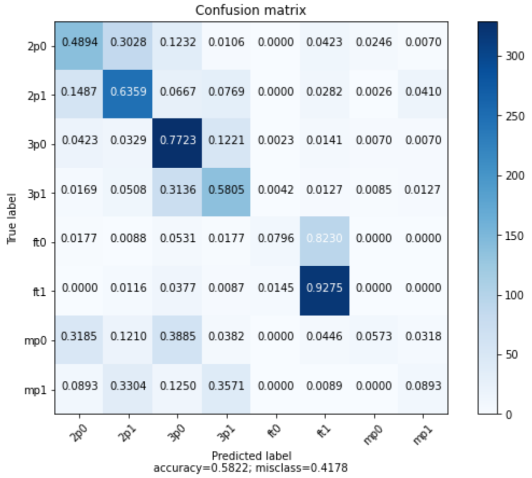


True label \ Predicted label	2p0	2p1	3p0	3p1	ft0	ft1	mp0	mp1
2p0	0.3803	0.4542	0.0880	0.0211	0.0000	0.0563	0.0000	0.0000
2p1	0.1744	0.7026	0.0282	0.0615	0.0026	0.0282	0.0000	0.0026
3p0	0.0610	0.1080	0.5211	0.2512	0.0000	0.0587	0.0000	0.0000
3p1	0.0466	0.2288	0.1907	0.5042	0.0000	0.0297	0.0000	0.0000
ft0	0.0088	0.0000	0.0265	0.0000	0.0088	0.9558	0.0000	0.0000
ft1	0.0029	0.0000	0.0000	0.0058	0.0000	0.9913	0.0000	0.0000
mp0	0.2611	0.2739	0.3312	0.0701	0.0000	0.0573	0.0064	0.0000
mp1	0.1339	0.3929	0.1607	0.2500	0.0000	0.0625	0.0000	0.0000

Classification Report

	precision	recall	f1-score	support
0	0.40	0.38	0.39	284
1	0.46	0.70	0.56	390
2	0.59	0.52	0.55	426
3	0.40	0.50	0.45	236
4	0.50	0.01	0.02	113
5	0.65	0.99	0.79	345
6	1.00	0.01	0.01	157
7	0.00	0.00	0.00	112
accuracy			0.52	2063
macro avg	0.50	0.39	0.35	2063
weighted avg	0.52	0.52	0.46	2063

Figure 5.13 *SD Confusion matrix plot and classification Report for class-8 Optical flow (grayscale) input video*

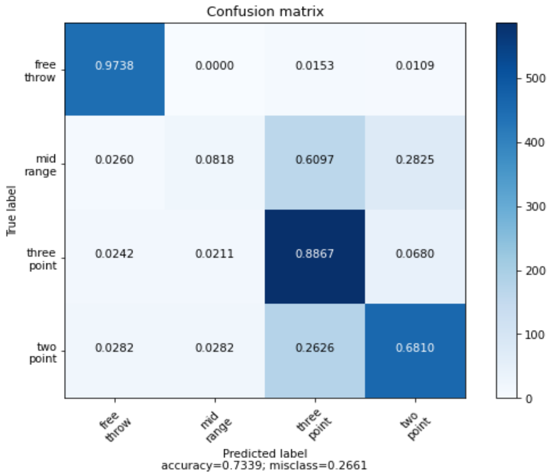


True label \ Predicted label	2p0	2p1	3p0	3p1	ft0	ft1	mp0	mp1
2p0	0.4894	0.3028	0.1232	0.0106	0.0000	0.0423	0.0246	0.0070
2p1	0.1487	0.6359	0.0667	0.0769	0.0000	0.0282	0.0026	0.0410
3p0	0.0423	0.0329	0.7723	0.1221	0.0023	0.0141	0.0070	0.0070
3p1	0.0169	0.0508	0.3136	0.5805	0.0042	0.0127	0.0085	0.0127
ft0	0.0177	0.0088	0.0531	0.0177	0.0796	0.8230	0.0000	0.0000
ft1	0.0000	0.0116	0.0377	0.0087	0.0145	0.9275	0.0000	0.0000
mp0	0.3185	0.1210	0.3885	0.0382	0.0000	0.0446	0.0573	0.0318
mp1	0.0893	0.3304	0.1250	0.3571	0.0000	0.0089	0.0000	0.0893

Classification Report

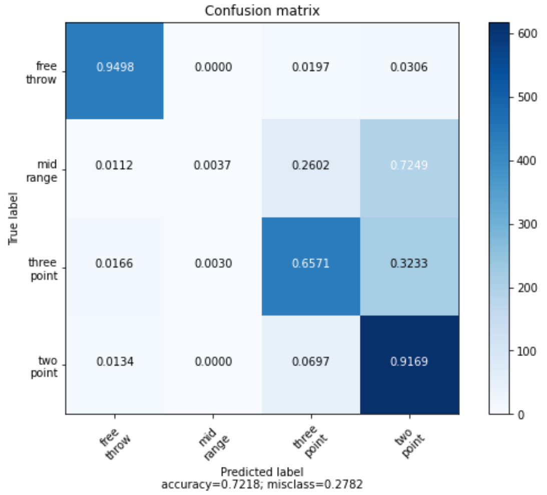
	precision	recall	f1-score	support
0	0.49	0.49	0.49	284
1	0.59	0.64	0.61	390
2	0.59	0.77	0.67	426
3	0.50	0.58	0.54	236
4	0.56	0.08	0.14	113
5	0.71	0.93	0.80	345
6	0.41	0.06	0.10	157
7	0.26	0.09	0.13	112
accuracy			0.58	2063
macro avg	0.51	0.45	0.44	2063
weighted avg	0.55	0.58	0.54	2063

Figure 5.14 *SD Confusion Matrix plot and classification Report for class-8 RGB Optical flow input video.*



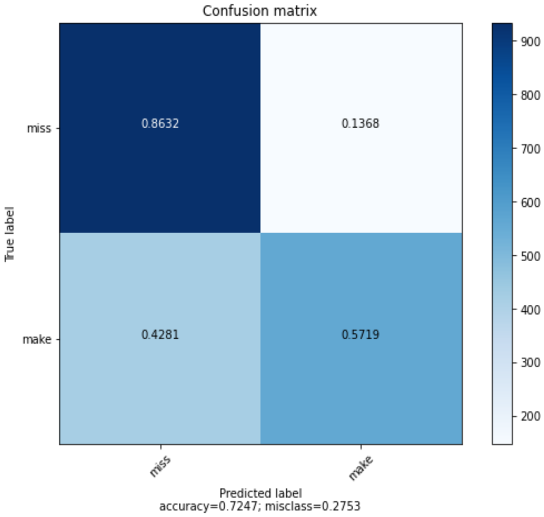
```
Classification Report

              precision    recall f1-score   support

           0       0.91      0.97      0.94       458
           1       0.40      0.08      0.14       269
           2       0.63      0.89      0.74       662
           3       0.78      0.68      0.73       674

    accuracy                           0.73      2063
   macro avg       0.68      0.66      0.64      2063
weighted avg       0.71      0.73      0.70      2063
```

Figure 5.15 *SD Confusion matrix plot and classification Report for class-4 Optical flow (grayscale) input video data*



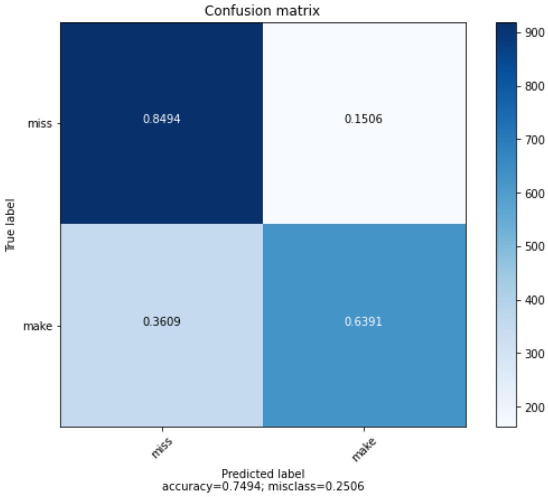
```
Classification Report

              precision    recall f1-score   support

           0       0.95      0.95      0.95       458
           1       0.33      0.00      0.01       269
           2       0.78      0.66      0.71       662
           3       0.59      0.92      0.72       674

    accuracy                           0.72      2063
   macro avg       0.66      0.63      0.60      2063
weighted avg       0.70      0.72      0.68      2063
```

Figure 5.16 *SD Confusion Matrix plot and classification Report for class-4 RGB Optical flow input video data*



```
Classification Report

          precision    recall f1-score   support

       0       0.69      0.86      0.77      1082
       1       0.79      0.57      0.66       981

accuracy                           0.72      2063
macro avg       0.74      0.72      0.72      2063
weighted avg    0.74      0.72      0.72      2063
```

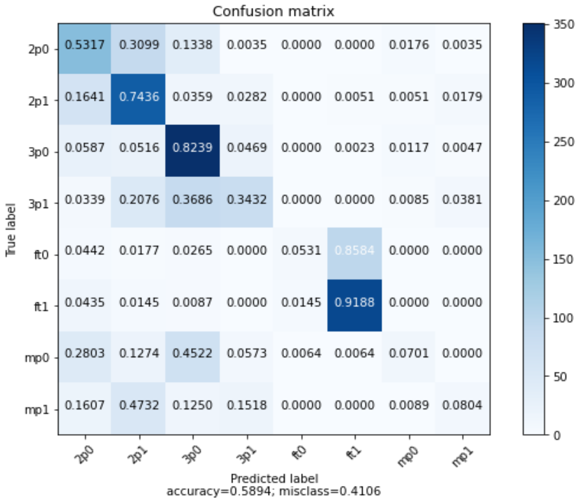
Figure 5.17 *SD Confusion matrix plot and classification Report for class-2 Optical flow(grayscale) input video data*



```
Classification Report

          precision recall  f1-score   support

       0       0.72      0.85      0.78      1082
       1       0.79      0.64      0.71       981

accuracy                           0.75      2063
macro avg       0.76      0.74      0.74      2063
weighted avg    0.76      0.75      0.75      2063
```

Figure 5.18 *SD Confusion Matrix plot and classification Report for class-2 RGB Optical flow input video data*

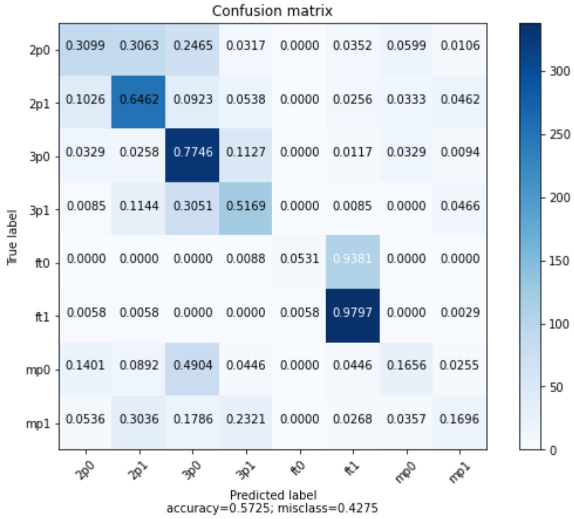
Figure 5.13 to Figure 5.18 represents the confusion matrix plot and classification report for RGB and grayscale optical flow videos for different groups. Here also the mid-range has been highly misclassified as three-point and two-point shots and have very low F1 score compared to other activity in the all the groups.



	2p0	2p1	3p0	3p1	ft0	ft1	mp0	mp1
2p0	0.5317	0.3099	0.1338	0.0035	0.0000	0.0000	0.0176	0.0035
2p1	0.1641	0.7436	0.0359	0.0282	0.0000	0.0051	0.0051	0.0179
3p0	0.0587	0.0516	0.8239	0.0469	0.0000	0.0023	0.0117	0.0047
3p1	0.0339	0.2076	0.3686	0.3432	0.0000	0.0000	0.0085	0.0381
ft0	0.0442	0.0177	0.0265	0.0000	0.0531	0.8584	0.0000	0.0000
ft1	0.0435	0.0145	0.0087	0.0000	0.0145	0.9188	0.0000	0.0000
mp0	0.2803	0.1274	0.4522	0.0573	0.0064	0.0064	0.0701	0.0000
mp1	0.1607	0.4732	0.1250	0.1518	0.0000	0.0000	0.0089	0.0804

```
                Classification Report

                 precision recall f1-score support

          0         0.46     0.53     0.49      284
          1         0.55     0.74     0.63      390
          2         0.60     0.82     0.70      426
          3         0.58     0.34     0.43      236
          4         0.50     0.05     0.10      113
          5         0.76     0.92     0.83      345
          6         0.42     0.07     0.12      157
          7         0.32     0.08     0.13      112

      accuracy                          0.59     2063
     macro avg      0.52     0.45     0.43     2063
  weighted avg      0.56     0.59     0.54     2063
```

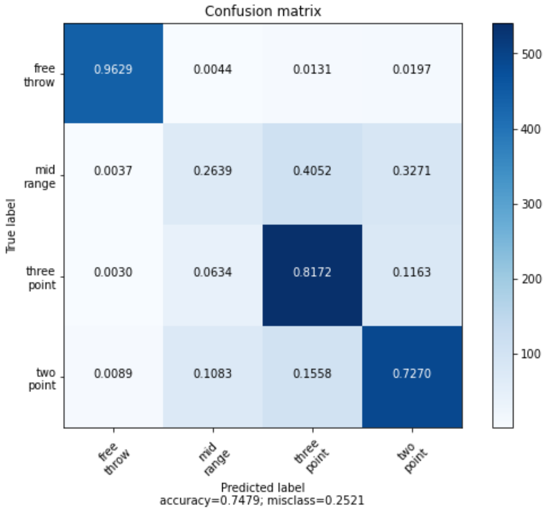
Figure 5.19 *SD Confusion matrix plot and classification Report for class-8 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video*



	2p0	2p1	3p0	3p1	ft0	ft1	mp0	mp1
2p0	0.3099	0.3063	0.2465	0.0317	0.0000	0.0352	0.0599	0.0106
2p1	0.1026	0.6462	0.0923	0.0538	0.0000	0.0256	0.0333	0.0462
3p0	0.0329	0.0258	0.7746	0.1127	0.0000	0.0117	0.0329	0.0094
3p1	0.0085	0.1144	0.3051	0.5169	0.0000	0.0085	0.0000	0.0466
ft0	0.0000	0.0000	0.0000	0.0088	0.0531	0.9381	0.0000	0.0000
ft1	0.0058	0.0058	0.0000	0.0000	0.0058	0.9797	0.0000	0.0029
mp0	0.1401	0.0892	0.4904	0.0446	0.0000	0.0446	0.1656	0.0255
mp1	0.0536	0.3036	0.1786	0.2321	0.0000	0.0268	0.0357	0.1696

```
                Classification Report

               precision recall  f1-score support

           0     0.51     0.31     0.38      284
           1     0.59     0.65     0.62      390
           2     0.55     0.77     0.64      426
           3     0.52     0.52     0.52      236
           4     0.75     0.05     0.10      113
           5     0.70     0.98     0.82      345
           6     0.35     0.17     0.23      157
           7     0.32     0.17     0.22      112

         accuracy                   0.57     2063
     macro avg   0.54     0.45     0.44     2063
  weighted avg   0.56     0.57     0.53     2063
```

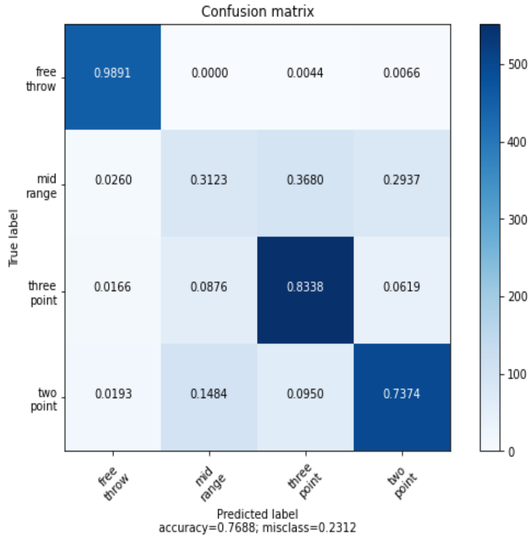
Figure 5.20 *SD Confusion matrix plot and classification Report for class-8 two-stream 3D CNN network using RGB and Optical flow input video.*



Classification Report

	precision	recall	f1-score	support
0	0.98	0.96	0.97	458
1	0.38	0.26	0.31	269
2	0.71	0.82	0.76	662
3	0.74	0.73	0.73	674
accuracy			0.75	2063
macro avg	0.70	0.69	0.69	2063
weighted avg	0.74	0.75	0.74	2063

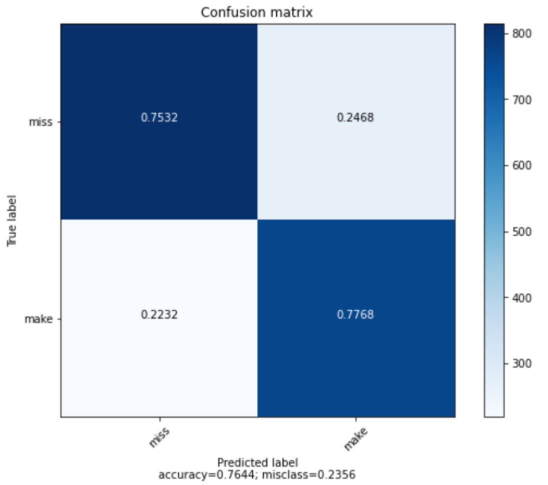
Figure 5.21 *SD Confusion matrix plot and classification Report for class-4 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video*



Classification Report

	precision	recall	f1-score	support
0	0.94	0.99	0.96	458
1	0.35	0.31	0.33	269
2	0.77	0.83	0.80	662
3	0.80	0.74	0.77	674
accuracy			0.77	2063
macro avg	0.71	0.72	0.71	2063
weighted avg	0.76	0.77	0.76	2063

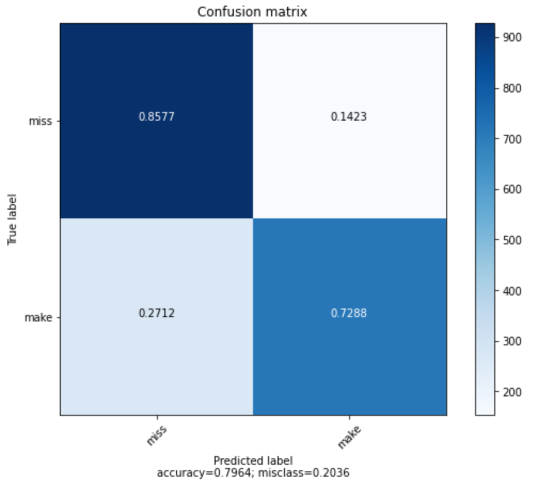
Figure 5.22 *SD Confusion matrix plot and classification Report for class-4 two-stream 3D CNN networks using RGB and Optical flow input video.*



Classification Report

	precision	recall	f1-score	support
0	0.79	0.75	0.77	1082
1	0.74	0.78	0.76	981
accuracy			0.76	2063
macro avg	0.76	0.76	0.76	2063
weighted avg	0.77	0.76	0.76	2063

Figure 5.23 *SD Confusion matrix plot and classification Report for class-2 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video*



Classification Report

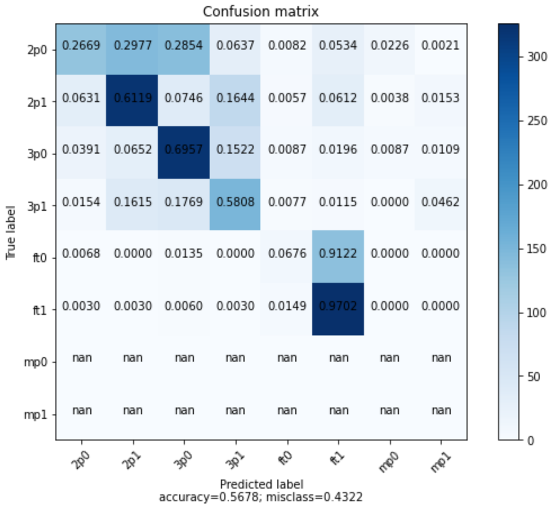
	precision	recall	f1-score	support
0	0.78	0.86	0.82	1082
1	0.82	0.73	0.77	981
accuracy			0.80	2063
macro avg	0.80	0.79	0.79	2063
weighted avg	0.80	0.80	0.80	2063

Figure 5.24 *SD Confusion matrix plot and classification Report for class-2 two-stream 3D CNN networks using RGB and Optical flow input video.*

Figure 5.19 to Figure 5.24 represents the confusion matrix and classification report for the two-stream CNN network. Here also we can see that in most cases RGB input data has higher accuracy than grayscale input and mid-range has the lowest score in the range of 24% for class-8 and 33% for class-4. This shows that it has been highly misclassified. For class -4, free-throw has the highest accuracy which is above 95% in all the input data.

5.5.2 Subject independent (SI)

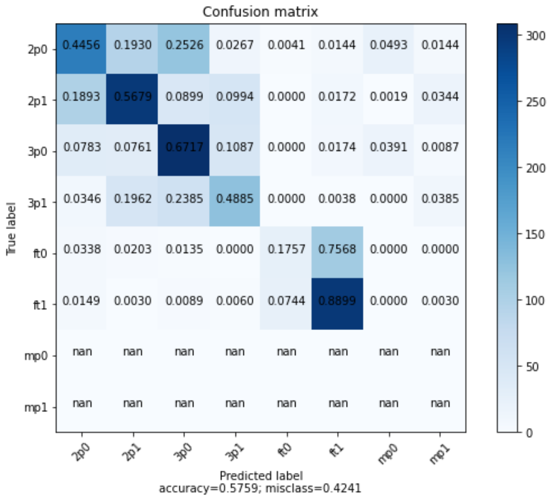
For subject independent analysis we use clips from 40 videos as training and 11 videos as testing data.



	2p0	2p1	3p0	3p1	ft0	ft1	mp0	mp1
2p0	0.2669	0.2977	0.2854	0.0637	0.0082	0.0534	0.0226	0.0021
2p1	0.0631	0.6119	0.0746	0.1644	0.0057	0.0612	0.0038	0.0153
3p0	0.0391	0.0652	0.6957	0.1522	0.0087	0.0196	0.0087	0.0109
3p1	0.0154	0.1615	0.1769	0.5808	0.0077	0.0115	0.0000	0.0462
ft0	0.0068	0.0000	0.0135	0.0000	0.0676	0.9122	0.0000	0.0000
ft1	0.0030	0.0030	0.0060	0.0030	0.0149	0.9702	0.0000	0.0000
mp0	nan	nan	nan	nan	nan	nan	nan	nan
mp1	nan	nan	nan	nan	nan	nan	nan	nan

```
Classification Report
              precision    recall  f1-score  support

           0       0.70      0.27      0.39      487
           1       0.59      0.61      0.60      523
           2       0.58      0.70      0.63      460
           3       0.45      0.58      0.50      260
           4       0.36      0.07      0.11      148
           5       0.61      0.97      0.75      336
           6       0.00      0.00      0.00        0
           7       0.00      0.00      0.00        0

    accuracy                           0.57     2214
   macro avg       0.41      0.40      0.37     2214
weighted avg       0.58      0.57      0.54     2214
```

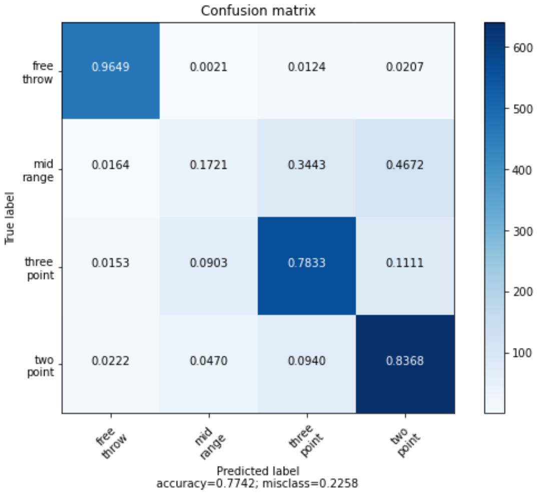
Figure 5.25 *SI Confusion matrix plot and classification Report for class-8 RGB (grayscale) input video*



	2p0	2p1	3p0	3p1	ft0	ft1	mp0	mp1
2p0	0.4456	0.1930	0.2526	0.0267	0.0041	0.0144	0.0493	0.0144
2p1	0.1893	0.5679	0.0899	0.0994	0.0000	0.0172	0.0019	0.0344
3p0	0.0783	0.0761	0.6717	0.1087	0.0000	0.0174	0.0391	0.0087
3p1	0.0346	0.1962	0.2385	0.4885	0.0000	0.0038	0.0000	0.0385
ft0	0.0338	0.0203	0.0135	0.0000	0.1757	0.7568	0.0000	0.0000
ft1	0.0149	0.0030	0.0089	0.0060	0.0744	0.8899	0.0000	0.0030
mp0	nan	nan	nan	nan	nan	nan	nan	nan
mp1	nan	nan	nan	nan	nan	nan	nan	nan

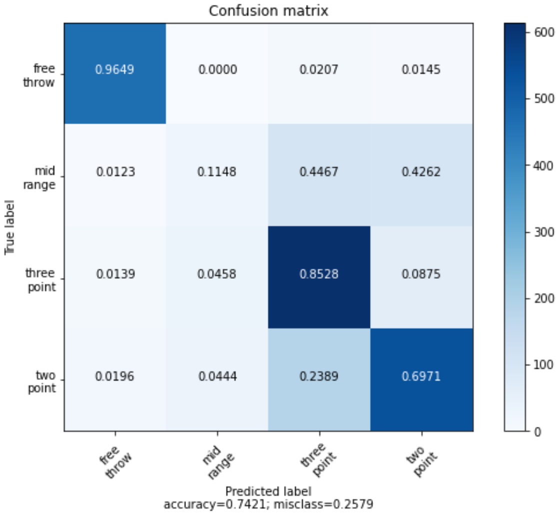
```
Classification Report
              precision    recall  f1-score  support

           0       0.58      0.45      0.51      487
           1       0.62      0.57      0.59      523
           2       0.57      0.67      0.61      460
           3       0.52      0.49      0.50      260
           4       0.49      0.18      0.26      148
           5       0.69      0.89      0.77      336
           6       0.00      0.00      0.00        0
           7       0.00      0.00      0.00        0

    accuracy                           0.58     2214
   macro avg       0.43      0.40      0.41     2214
weighted avg       0.59      0.58      0.57     2214
```

Figure 5.26 *SI Confusion matrix plot and classification Report for class-8 RGB input video*



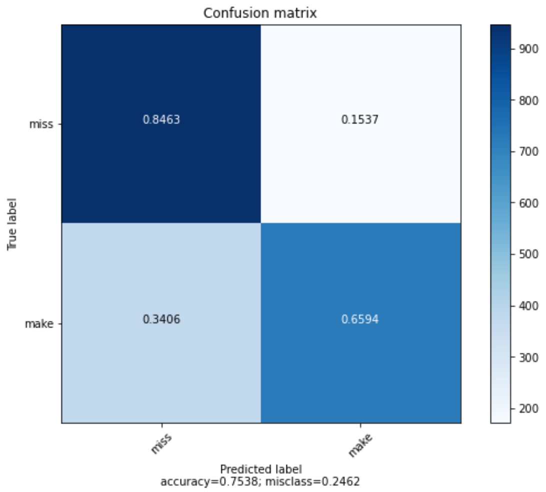
```
Classification Report

              precision    recall f1-score support

           0       0.94      0.96     0.95     484
           1       0.29      0.17     0.22     244
           2       0.78      0.78     0.78     720
           3       0.76      0.84     0.80     766

    accuracy                          0.77    2214
   macro avg       0.69      0.69     0.69    2214
weighted avg       0.75      0.77     0.76    2214
```

Figure 5.27 *SI Confusion matrix plot and classification Report for class-4 RGB (grayscale) input video*



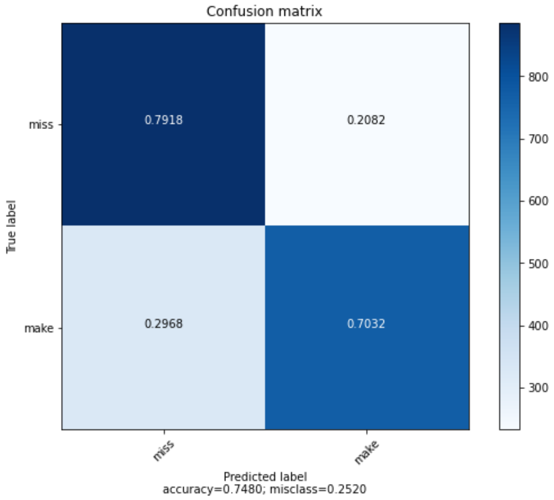
```
Classification Report

              precision    recall f1-score support

           0       0.94      0.96     0.95     484
           1       0.29      0.11     0.17     244
           2       0.67      0.85     0.75     720
           3       0.75      0.70     0.72     766

    accuracy                          0.74    2214
   macro avg       0.67      0.66     0.65    2214
weighted avg       0.72      0.74     0.72    2214
```

Figure 5.28 *SI Confusion matrix plot and classification Report for class-4 RGB input video*



```
Classification Report

              precision    recall f1-score support

           0       0.72      0.85      0.78     1119
           1       0.81      0.66      0.73     1095

    accuracy                           0.75     2214
   macro avg       0.76      0.75      0.75     2214
weighted avg       0.76      0.75      0.75     2214
```

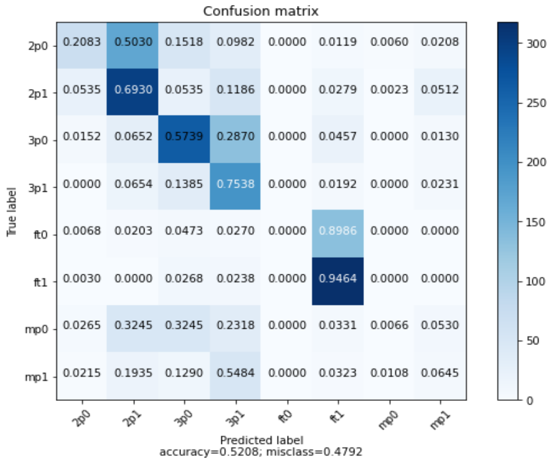
Figure 5.29 *SI Confusion matrix plot and classification Report for class-2 RGB (grayscale) input video*



```
Classification Report

              precision    recall f1-score support

           0       0.73      0.79      0.76     1119
           1       0.77      0.70      0.73     1095

    accuracy                           0.75     2214
   macro avg       0.75      0.75      0.75     2214
weighted avg       0.75      0.75      0.75     2214
```

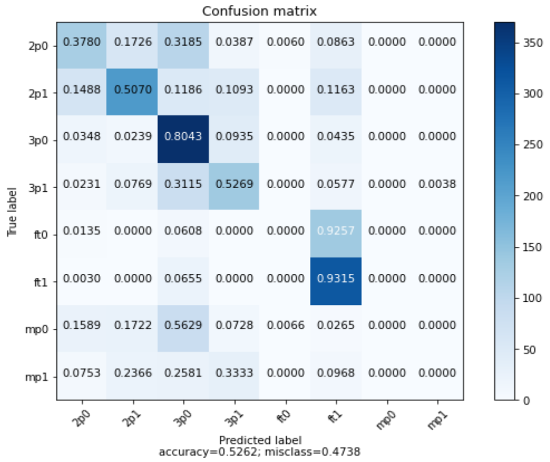
Figure 5.30 *SI Confusion matrix plot and classification Report for class-2 RGB input video*

Figure 5.25 to Figure 5.30 represents the confusion matrix plot and classification report for RGB video input for subject Independent classification types. The output is comparatively lower than the corresponding subject-dependent group.



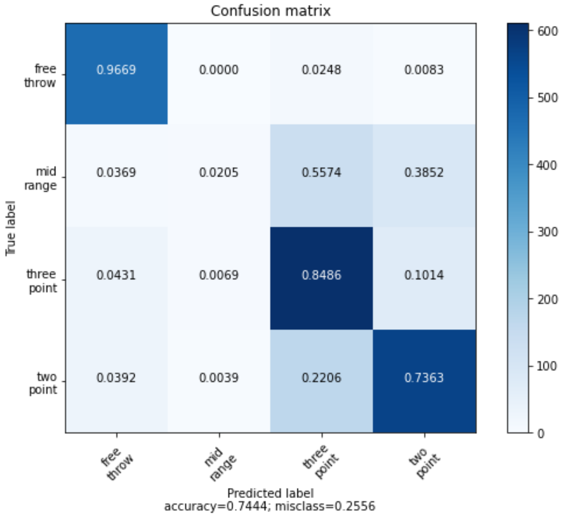
```
                Classification Report

              precision    recall f1-score support

           0       0.65      0.21     0.32      336
           1       0.51      0.69     0.59      430
           2       0.59      0.57     0.58      460
           3       0.38      0.75     0.51      260
           4       0.00      0.00     0.00      148
           5       0.63      0.95     0.76      336
           6       0.20      0.01     0.01      151
           7       0.11      0.06     0.08       93

    accuracy                          0.52     2214
   macro avg       0.38      0.41     0.36     2214
weighted avg       0.48      0.52     0.46     2214
```

Figure 5.31 *SI Confusion matrix plot and classification report for class-8 Optical flow(grayscale) input video*



```
                Classification Report

              precision    recall f1-score support

           0       0.51      0.38     0.44      336
           1       0.61      0.51     0.56      430
           2       0.49      0.80     0.61      460
           3       0.49      0.53     0.51      260
           4       0.00      0.00     0.00      148
           5       0.54      0.93     0.69      336
           6       0.00      0.00      0.00     151
           7       0.00      0.00     0.00       93

    accuracy                          0.53     2214
   macro avg       0.33      0.39     0.35     2214
weighted avg       0.44      0.53     0.46     2214
```

Figure 5.32 *SI Confusion matrix plot and classification report for class-8 Optical flow input video*

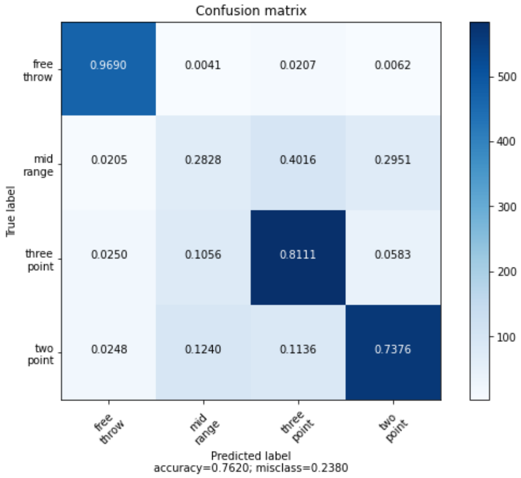


	free throw	mid range	three point	two point
free throw	0.9669	0.0000	0.0248	0.0083
mid range	0.0369	0.0205	0.5574	0.3852
three point	0.0431	0.0069	0.8486	0.1014
two point	0.0392	0.0039	0.2206	0.7363

Classification Report

	precision	recall	f1-score	support
0	0.87	0.97	0.92	484
1	0.38	0.02	0.04	244
2	0.66	0.85	0.74	720
3	0.77	0.74	0.75	766
accuracy			0.74	2214
macro avg	0.67	0.64	0.61	2214
weighted avg	0.71	0.74	0.71	2214

Figure 5.33 *SI Confusion matrix plot and classification report for class-4 Optical flow(grayscale) input video*

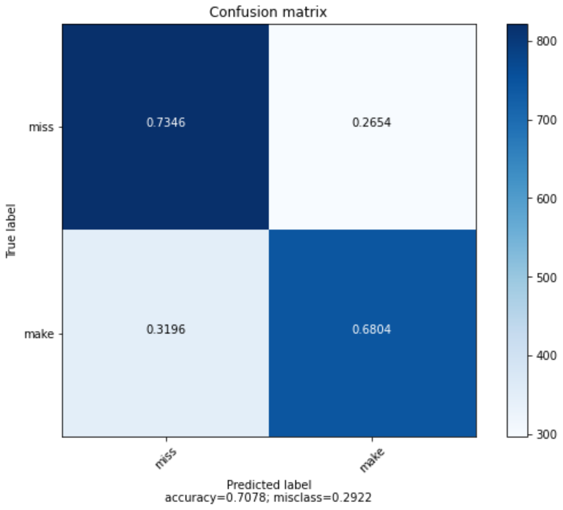


	free throw	mid range	three point	two point
free throw	0.9690	0.0041	0.0207	0.0062
mid range	0.0205	0.2828	0.4016	0.2951
three point	0.0250	0.1056	0.8111	0.0583
two point	0.0248	0.1240	0.1136	0.7376

Classification Report

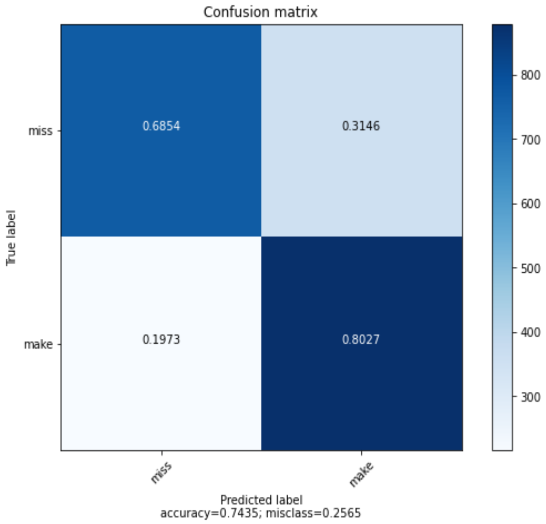
	precision	recall	f1-score	support
0	0.92	0.97	0.94	484
1	0.29	0.28	0.28	244
2	0.75	0.81	0.78	720
3	0.83	0.74	0.78	766
accuracy			0.76	2214
macro avg	0.70	0.70	0.70	2214
weighted avg	0.76	0.76	0.76	2214

Figure 5.34 *SI Confusion matrix plot and classification report for class-4 Optical flow input video*



```
                 Classification Report

                  precision recall f1-score support

         0         0.70     0.73     0.72      1119
         1         0.71     0.68     0.70      1095

   accuracy                          0.71      2214
  macro avg        0.71     0.71     0.71      2214
weighted avg       0.71     0.71     0.71      2214
```

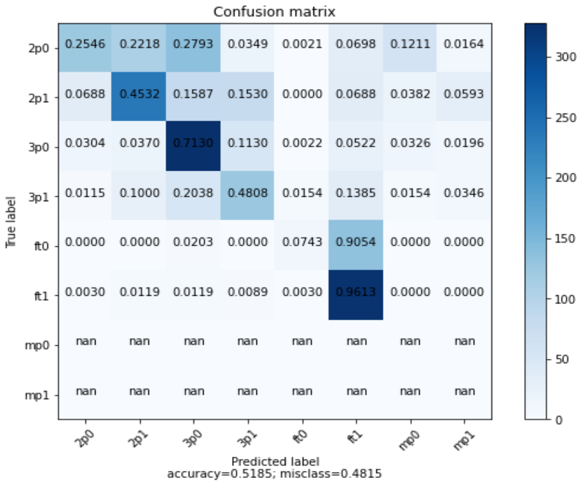
Figure 5.35 *SI Confusion matrix plot and classification report for class-2 Optical flow(grayscale) input video*



```
Classification Report

               precision   recall f1-score support

        0          0.78     0.69     0.73      1119
        1          0.71     0.80     0.76      1095

    accuracy                         0.74      2214
   macro avg       0.75     0.74     0.74      2214
 weighted avg      0.75     0.74     0.74       221
```

Figure 5.36 *SI Confusion matrix plot and classification report for class-2 Optical flow RGB input video*

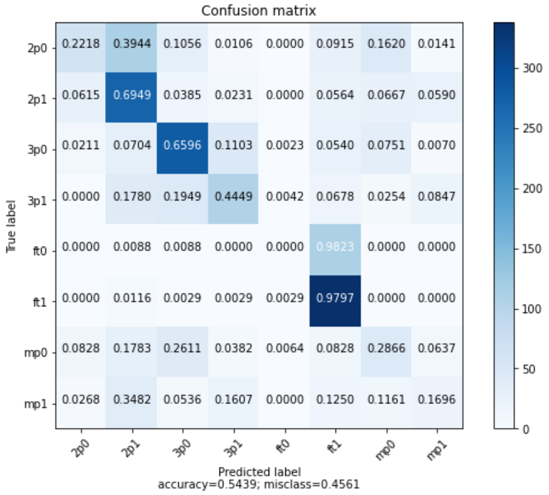
Figure 5.31 to figure 5.36 represents the confusion matrix plot and classification report for grayscale and RGB video of optical flow video input of subject independent classification type. Here, in all the groups, the accuracy is higher when using RGB optical input video than using grayscale input video.



	2p0	2p1	3p0	3p1	ft0	ft1	mp0	mp1
2p0	0.2546	0.2218	0.2793	0.0349	0.0021	0.0698	0.1211	0.0164
2p1	0.0688	0.4532	0.1587	0.1530	0.0000	0.0688	0.0382	0.0593
3p0	0.0304	0.0370	0.7130	0.1130	0.0022	0.0522	0.0326	0.0196
3p1	0.0115	0.1000	0.2038	0.4808	0.0154	0.1385	0.0154	0.0346
ft0	0.0000	0.0000	0.0203	0.0000	0.0743	0.9054	0.0000	0.0000
ft1	0.0030	0.0119	0.0119	0.0089	0.0030	0.9613	0.0000	0.0000
mp0	nan	nan	nan	nan	nan	nan	nan	nan
mp1	nan	nan	nan	nan	nan	nan	nan	nan

```
Classification Report

              precision    recall f1-score support

           0       0.70      0.25      0.37     487
           1       0.60      0.45      0.52     523
           2       0.54      0.71      0.61     460
           3       0.45      0.48      0.47     260
           4       0.61      0.07      0.13     148
           5       0.55      0.96      0.70     336
           6       0.00      0.00      0.00       0
           7       0.00      0.00      0.00       0

    accuracy                          0.52    2214
   macro avg       0.43      0.37     0.35    2214
weighted avg       0.59      0.52     0.50    2214
```

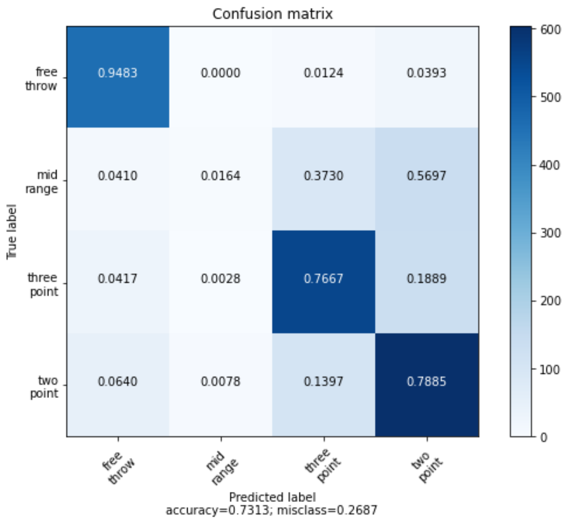
Figure 5.37 *SI Confusion matrix plot and classification report for class-8 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video.*



	2p0	2p1	3p0	3p1	ft0	ft1	mp0	mp1
2p0	0.2218	0.3944	0.1056	0.0106	0.0000	0.0915	0.1620	0.0141
2p1	0.0615	0.6949	0.0385	0.0231	0.0000	0.0564	0.0667	0.0590
3p0	0.0211	0.0704	0.6596	0.1103	0.0023	0.0540	0.0751	0.0070
3p1	0.0000	0.1780	0.1949	0.4449	0.0042	0.0678	0.0254	0.0847
ft0	0.0000	0.0088	0.0088	0.0000	0.0000	0.9823	0.0000	0.0000
ft1	0.0000	0.0116	0.0029	0.0029	0.0029	0.9797	0.0000	0.0000
mp0	0.0828	0.1783	0.2611	0.0382	0.0064	0.0828	0.2866	0.0637
mp1	0.0268	0.3482	0.0536	0.1607	0.0000	0.1250	0.1161	0.1696

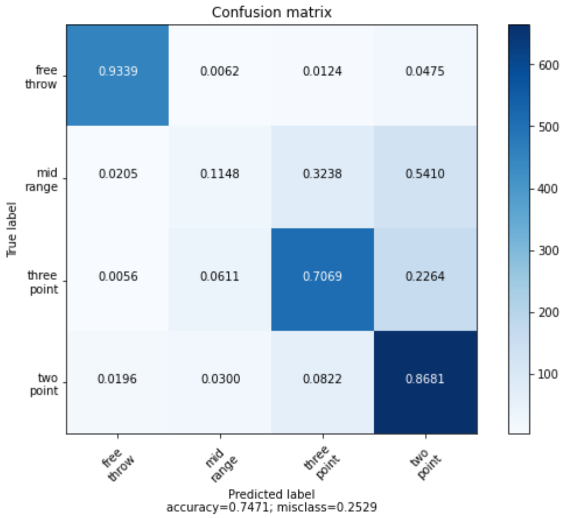
```
              precision  recall  f1-score  support

           0       0.56    0.22     0.32      284
           1       0.51    0.69     0.59      390
           2       0.67    0.66     0.66      426
           3       0.56    0.44     0.49      236
           4       0.00    0.00     0.00      113
           5       0.60    0.98     0.74      345
           6       0.27    0.29     0.28      157
           7       0.24    0.17     0.20      112

    accuracy                        0.54     2063
   macro avg       0.43    0.43     0.41     2063
weighted avg       0.51    0.54     0.51     2063
```

Figure 5.38 *SI Confusion matrix plot and classification report for class-8 two-stream 3D CNN networks using RGB and Optical flow input video.*



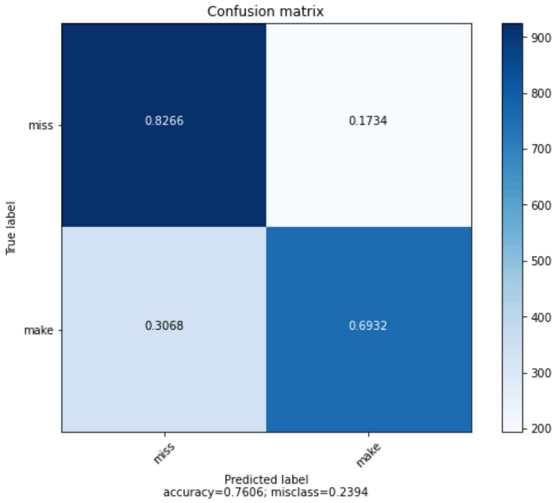
```
Classification Report

             precision  recall  f1-score  support

           0    0.84     0.95     0.89      484
           1    0.33     0.02     0.03      244

           2    0.73     0.77     0.75      720
           3    0.67     0.79     0.73      766

    accuracy                      0.73     2214
   macro avg    0.64     0.63     0.60     2214
weighted avg    0.69     0.73      0.69    2214
```

Figure 5.39 *SI Confusion matrix plot and classification report for class-4 two-stream 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video.*



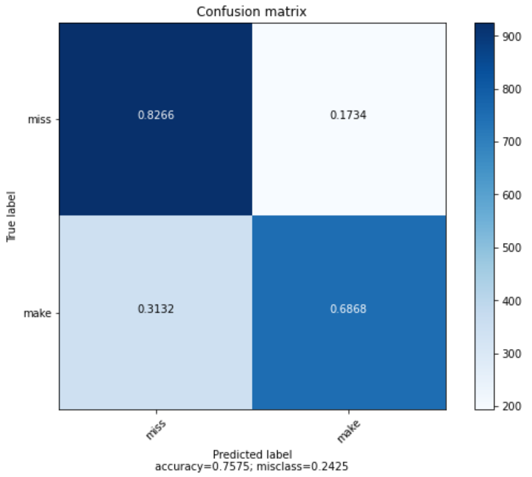
```
Classification Report

              precision   recall f1-score support

           0     0.95      0.93     0.94      484
           1     0.29      0.11     0.16      244
           2     0.77      0.71     0.74      720
           3     0.68      0.87     0.76      766

    accuracy                        0.75     2214
   macro avg     0.67      0.66     0.65     2214
weighted avg     0.73      0.75     0.73     2214
```

Figure 5.40 *SI Confusion matrix plot and classification report for class-4 two stream CNN networks using RGB and Optical flow input video.*



```
                 Classification Report

              precision recall f1-score support

           0       0.73    0.83    0.78    1119
           1       0.80    0.69    0.74    1095

    accuracy                       0.76    2214
   macro avg       0.76    0.76    0.76    2214
weighted avg       0.76    0.76    0.76    2214
```

Figure 5.41 *SI Confusion matrix plot and classification report for class-2 two streams 3D CNN networks using RGB (grayscale) and Optical flow(grayscale) input video.*



```
                 Classification Report

              precision recall f1-score support

           0       0.73    0.83    0.78    1119
           1       0.79    0.69    0.74    1095

    accuracy                       0.76    2214
   macro avg       0.76    0.76    0.76    2214
weighted avg       0.76    0.76    0.76    2214
```

Figure 5.42 *SI Confusion matrix plot and classification report for class-2 two-stream 3D CNN networks using RGB and Optical flow input video.*

From figure 5.37 to Figure 5.42, which represents the confusion matrix plot and classification report for Subject Independent two-stream CNN network for all the groups. Here also the mid-range has been highly misclassified as two-point and three-point shots and all other output is consistent with other previous outputs.

Table 5.1 shows the overall accuracy table for all our experimental setups. From the table for the 8-class classification, the highest accuracy we got is from the 3D convnet using grayscale RGB data which is around 59.5%. using color optical flow video as input has highly improved the performance for a class-8 group where there is not much difference in other groups. For class 4, the highest accuracy is for 3D Convnet using input from RGB video data which is around 80%. The performance is highest for the free throw which has higher precision and recall compared to other class activities. Here also mid-range has been highly misclassified as two points and three-point. For the class 2 group, the highest is by using two-stream 3D Convnet which is around 76%. In all cases, RGB videos have higher performance than optical flow video. The two-stream network hasn’t shown significant improvement except for class-2 subject independent analysis. Among different groups, the model has higher accuracy in determining the range of the shot taken which is class-4 group except for using a two-stream 3D convnet for classifying the make and miss of the shots. Subject-dependent analysis has higher performance than subject independent analysis in almost all the cases.

Table 5.1 *Accuracy for all the experiments with three models used.*

Methods	**Accuracy**					
	Subject Dependent					
	8-class		**4-class**		**2-class**	
	1D	3D	1D	3D	1D	3D
3D ConvNet	**59.48%**	59.19%	79.50%	**80.27%**	77.31%	**79.59%**
3D optical ConvNet	51.72%	58.22%	73.39%	72.18%	72.47%	74.94%
Two stream 3D ConvNet	58.94%	57.25%	74.79%	76.88%	76.44%	79.64%
	Accuracy					
	Subject Independent					
	8-class		**4-class**		**2-class**	
	1D	3D	1D	3D	1D	3D
3D ConvNet	56.78%	57.59%	77.42%	74.21%	75.38%	74.80%
3D optical ConvNet	52.08%	52.62%	74.44%	76.20%	70.78%	74.35%
Two stream 3D ConvNet	51.85%	54.39%	73.13%	74.71%	**76.06%**	75.75%

5.6 Discussion and Conclusions

From the analysis, we can see that for classifying different activities, mid-range shots have been highly misclassified as three-point shots and two-point shots. This can be due to the smaller number of training samples for mid-range shots compared to the other class labels. Also, as mid-range shots are very much similar to the two-point shots, the model couldn’t be able to learn more specific features from the training data. More training data for mid-range shots should be added with deeper network training that can improve the results. Also, the free throw has a comparatively higher performance than other groups. This can be because the camera angle, the player movement, the dynamic nature of the background is highly stable, or constant compared to the video while performing other activities. Because of the highly similar interclass data, high mobility of the players, coaches, and audience during the game, the dynamic nature of background due to the movement of the camera, and availability of low training data due to the high requirement of memory space for video data have been some of the reasons for the low classification accuracy. Eliminating these difficulties can highly improve classification performance.

In this study, we developed a video dataset for basketball to classify different scoring activities based on different criteria using only the video clips broadcasted from the broadcasting media. The objective of this dataset is to provide a dataset collected from easily available video data without the use of any special camera and camera setup. This study can also help for classifying different activities relates to basketball for developing an automated activity recognition system for score updating, foul play detection, assisting decision making for the referee, and other activities accurately and in real-time. This can also be the future work for this research.

Other future works include adding more classes or activities to the dataset such as dribbling, moving forward with the ball, foul play, time out, and other specific activities that happen during the play. Also, removing the challenges discussed such as background removal, ball detection, can be implemented for improving the recognition accuracy. Integrating ball tracking system and player tracking system can be used for better performance along with using higher-dimensional data adding more training data should be able to increase the performance of the model.

CHAPTER VI – FUTURE WORK

6.1 Future work

This section highlights some of the future works that can be extended from the proposed methods and dataset.

With recent development in sensors and sensor technology, new wearable devices which can be very comfortable and acceptable by sports personnel to wear during the game can be developed. The data collected from these sensors can represent a wide range of activities performed by the players with stable recognition. This can help to gather more data for different activities and can help to gather data from real-life activities rather than lab setup and monitored activities performance.

The research focuses mainly on single viewpoint recognition frames collected from a single device. Multi viewpoint acquisition can be considered for training that has been captured by multiple devices for a similar activity to be a more realistic approach.

Also, deep models for training the training data can be used to extract features that can help in classifying close inter-class classification. Also, a multi-label classification system and model can be developed that can classify multiple actions of humans that are more applicable to the real world. The improvement in the collection and maintenance of the quality dataset so that it can provide adequate information for the model to learn and the improvement in the training model that can classify different activities taking the challenges it faces during vision-based activity recognition can be two sectors for future development.

6.2 Limitations/Challenges

In recent years, many research achievements have been achieved in the field of HAR in different fields such as sports, video surveillance, movies, and healthcare. Different methods have been used in activity recognition. Using sensors and video data for HAR can have many limitations which mainly depend on the devices, collected data quality, experimental environments, light variations, moving background[144], change in perspective, occlusion of another object, noise, etc. [145][146]. Here we present some of the challenges and limitations in the field of HAR categorized on different topics.

Dataset: Video datasets are high memory required datasets. The prospect of loading the entire dataset into the local memory is impractical. Some solutions can be to use a URL passing library to dynamically download the videos from their YouTube links and overwrite the videos currently in memory. A parallel computing system is used such that these batch can be loaded and preprocessed on a separate machine than the one which is training the model. Also, the lack of a standard benchmark dataset that gives all the information about different scenarios and actions for real-life representation is missing. This has reduced the effective evaluation and training for HAR. More efforts should be needed to prepare a quality datasets with quality information and uniform protocol should be applied for quantitative comparison on those benchmark datasets[109].

Intraclass variation and Inter-class similarity: In HAR, one of the challenges is the intraclass variation and inter-class similarity. The same activity can differ between different subjects based on body size, clothing, personality, and other factors. For example, the way a person walk can be different and unique for everyone. Also, different activities may look similar like walking and jogging can be very similar to running. Also, multiple actions such as drinking tea while talking on the phone can be hard to train which adds more challenges in the recognition process [147]. For analyzing activities in the crowd[148], or places, where the subject number is high, can add challenges in recognizing activities[149]. These overlapping actions can bring uncertainty in the recognition process[150]. A deep learning model can be developed that can distinguish these actions by learning unique features with more detailed data for these actions and activities and a more multileveled prediction model can be trained with hybrid devices for composite activities recognition.

Complex and varying backgrounds: Most real-world videos contain dynamic background, occlusions, illumination variance, noise, different lighting, a different viewpoint, low-quality images/videos. In sensor-based, the presence of noise, unwanted signal detection, and low-quality sensors and transmission systems can affect the quality of the data being collected. These issues add more challenges and complexity to HAR. The use of multi-modal technology using different sources of data such as RGB, depth, a skeleton in the video, and multi-sensors data in sensor-based HAR can be used [151].

Real-time Analysis: HAR system can be more resource-demanding and energy-consuming. Most application such as video surveillance, elderly care which need real-time accurate sensing. With large sensors and video data, the computing complexity will increase with the need for more computing power, energy memory. Also, some applications such as security should be able to predict certain activities of a person by analyzing certain behavior and be able to stop future unwanted movement. These requirements have added more challenges in real-time automatic activity recognition. More study should be focused on reducing the resources requirement developing energy-efficient system[152] by designing with lower sampling frequency, deep transfer learning, and deeper model[153] with new methods. The challenge will be is to minimize the overall operational cost by reducing the computing and bandwidth resources.

Privacy: With wide application in surveillance, elderly assistance, and monitoring systems, the HAR system has the challenge of balancing between monitoring and violating the intimacy and privacy of the person. Installation of devices for monitoring the activity at home can arise privacy constraints with misuse of data collected from these devices. Also, the reliability, integrity, and security of collected data and information have added more challenges in this field. The use of devices such as a smartphone for monitoring can help in privacy constraints of the data as the control of these devices will be in the hand of the users. Building more secure and acceptable devices and the system can help people to be more comfortable with this technology and devices and will be able to overcome these challenges.

6.3 Conclusion

HAR has become an integral part of analyzing and interpreting human activities in different applications of computer vision, robotics, and many more. With data collected from different easily available sensors such as accelerometer, gyroscope, magnetometer embedded in handheld devices such as smartphones, smartwatches have made data collection easy. In the early part of the research, we compare the performance of these data collected from the accelerometer of a smartphone placed at a different position on the body. With different approaches such as machine learning and deep learning architecture, we analyze the data collected from the sensors and classify human activities. The study found that with the use of multiple sensors and a more balanced dataset used for training the model, higher accuracy can be achieved using only the raw data coming out from the sensors. With sensors, as it can have some limitations applying in some fields such as sports, we used video-based activity recognition based on sports data for activity recognition and classification. For this, we used state of an art approach with a pre-trained model in three benchmark datasets related to sports. The results show relatively comparable results.

Apart from classifying different sports activities based on activity performed by the person in the video, we developed our dataset related to basketball to classify scoring activities related to the single sports. For that, we collected videos of basketball games that have been broadcasted in different broadcasting media, prepared a labeled dataset consisting of video clips of different actions, and used them to classify different activities based on different criteria. The results have provided the baseline output performance and the challenging dataset will be helpful for researchers to classify activities in computer vision having very similar intraclass properties.

REFERENCES

[1] C. Dhiman and D. K. Vishwakarma, “A Robust Framework for Abnormal Human Action Recognition Using $\boldsymbol{\mathcal{R}}$ -Transform and Zernike Moments in Depth Videos,” *IEEE Sens. J.*, vol. 19, no. 13, pp. 5195–5203, 2019, doi: 10.1109/JSEN.2019.2903645.

[2] P. Corbishley and E. Rodriguez-Villegas, “Breathing Detection: Towards a Miniaturized, Wearable, Battery-Operated Monitoring System,” *IEEE Trans. Biomed. Eng.*, vol. 55, no. 1, pp. 196–204, 2008, doi: 10.1109/TBME.2007.910679.

[3] Z. Wang, Z. Yang, and T. Dong, “A Review of Wearable Technologies for Elderly Care that Can Accurately Track Indoor Position, Recognize Physical Activities and Monitor Vital Signs in Real Time,” *Sensors*, vol. 17, no. 2, 2017.

[4] A. Hölzemann and K. Van Laerhoven, “Using Wrist-Worn Activity Recognition for Basketball Game Analysis,” in *Proceedings of the 5th International Workshop on Sensor-based Activity Recognition and Interaction*, 2018, pp. 13:1--13:6, doi: 10.1145/3266157.3266217.

[5] W.-L. Lu, J.-A. Ting, J. J. Little, and K. P. Murphy, “Learning to track and identify players from broadcast sports videos,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1704–1716, 2013.

[6] W. Chi, J. Wang, and M. Q.-H. Meng, “A Gait Recognition Method for Human Following in Service Robots,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 48, no. 9, pp. 1429–1440, 2018, doi: 10.1109/TSMC.2017.2660547.

[7] J. Kang, R. Yu, X. Huang, S. Maharjan, Y. Zhang, and E. Hossain, “Enabling

Localized Peer-to-Peer Electricity Trading Among Plug-in Hybrid Electric Vehicles Using Consortium Blockchains," *IEEE Trans. Ind. Informatics*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017, doi: 10.1109/TII.2017.2709784.

[8] S. Samanta and B. Chanda, "Space-Time Facet Model for Human Activity Classification," *IEEE Trans. Multimed.*, vol. 16, no. 6, pp. 1525–1535, 2014, doi: 10.1109/TMM.2014.2326734.

[9] H. Tabatabaee Malazi and M. Davari, "Combining emerging patterns with random forest for complex activity recognition in smart homes," *Appl. Intell.*, vol. 48, no. 2, pp. 315–330, 2018, doi: 10.1007/s10489-017-0976-2.

[10] L. Schrader *et al.*, "Advanced Sensing and Human Activity Recognition in Early Intervention and Rehabilitation of Elderly People," *J. Popul. Ageing*, vol. 13, no. 2, pp. 139–165, 2020, doi: 10.1007/s12062-020-09260-z.

[11] L. Lonini *et al.*, "Wearable sensors for Parkinson's disease: which data are worth collecting for training symptom detection models," *npj Digit. Med.*, vol. 1, no. 1, p. 64, 2018, doi: 10.1038/s41746-018-0071-z.

[12] M. Bachlin *et al.*, "Wearable Assistant for Parkinson's Disease Patients With the Freezing of Gait Symptom," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 436–446, Mar. 2010, doi: 10.1109/TITB.2009.2036165.

[13] H. Gjoreski, M. Lustrek, and M. Gams, "Accelerometer Placement for Posture Recognition and Fall Detection," in *2011 Seventh International Conference on Intelligent Environments*, 2011, pp. 47–54, doi: 10.1109/IE.2011.11.

[14] P. Dreuw *et al.*, "The signspeak project - Bridging the gap between signers and speakers," *J. Speech Lang. Hear. Res. - J SPEECH LANG Hear RES*, 2010.

[15] V. Hernandez, T. Suzuki, and G. Venture, “Convolutional and recurrent neural network for human activity recognition: Application on American sign language,” *PLoS One*, vol. 15, no. 2, pp. 1–12, 2020, doi: 10.1371/journal.pone.0228869.

[16] A. Ullah, K. Muhammad, I. U. Haq, and S. W. Baik, “Action recognition using optimized deep autoencoder and CNN for surveillance data streams of non-stationary environments,” *Futur. Gener. Comput. Syst.*, vol. 96, pp. 386–397, 2019, doi: https://doi.org/10.1016/j.future.2019.01.029.

[17] R. M. M. Vallim, J. A. Andrade Filho, R. F. de Mello, and A. C. P. L. F. de Carvalho, “Online behavior change detection in computer games,” *Expert Syst. Appl.*, vol. 40, no. 16, pp. 6258–6265, 2013, doi: https://doi.org/10.1016/j.eswa.2013.05.059.

[18] L. Piyathilaka and S. Kodagoda, “Human Activity Recognition for Domestic Robots BT - Field and Service Robotics: Results of the 9th International Conference,” L. Mejias, P. Corke, and J. Roberts, Eds. Cham: Springer International Publishing, 2015, pp. 395–408.

[19] F. Fusier *et al.*, “Video understanding for complex activity recognition,” *Mach. Vis. Appl.*, vol. 18, pp. 167–188, 2006.

[20] S. Wang and G. Zhou, “A review on radio based activity recognition,” *Digit. Commun. Networks*, vol. 1, no. 1, pp. 20–29, 2015, doi: https://doi.org/10.1016/j.dcan.2015.02.006.

[21] X. Qi, G. Zhou, Y. Li, and G. Peng, “RadioSense: Exploiting Wireless Communication Patterns for Body Sensor Network Activity Recognition,” in *2012 IEEE 33rd Real-Time Systems Symposium*, 2012, pp. 95–104, doi:

10.1109/RTSS.2012.62.

[22] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, “E-Eyes: Device-Free Location-Oriented Activity Identification Using Fine-Grained WiFi Signatures,” in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014, pp. 617–628, doi: 10.1145/2639108.2639143.

[23] B. Kellogg, V. Talla, and S. Gollakota, “Bringing Gesture Recognition to All Devices,” in *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*, Apr. 2014, pp. 303–316, [Online]. Available: https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/kellogg.

[24] M. Scholz *et al.*, “SenseWaves: Radiowaves for context recognition,” *Video Submiss. Pervasive’11*, 2011.

[25] Z. Hussain, M. Sheng, and W. E. Zhang, “Different Approaches for Human Activity Recognition: {A} Survey,” *CoRR*, vol. abs/1906.0, 2019, [Online]. Available: http://arxiv.org/abs/1906.05074.

[26] H. Zheng and X.-M. Zhang, “A Cross-Modal Learning Approach for Recognizing Human Actions,” *IEEE Syst. J.*, pp. 1–9, 2020, doi: 10.1109/JSYST.2020.3001680.

[27] V. Magnanimo, M. Saveriano, S. Rossi, and D. Lee, “A Bayesian approach for task recognition and future human activity prediction,” in *The 23rd IEEE International Symposium on Robot and Human Interactive Communication*, 2014, pp. 726–731, doi: 10.1109/ROMAN.2014.6926339.

[28] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, “Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the

recognition of human actions," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 1932–1939, doi: 10.1109/CVPR.2009.5206821.

[29] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893 vol. 1, doi: 10.1109/CVPR.2005.177.

[30] L. Chen, C. D. Nugent, and H. Wang, "A Knowledge-Driven Approach to Activity Recognition in Smart Homes," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 961–974, 2012, doi: 10.1109/TKDE.2011.51.

[31] Z. Zeng and Q. Ji, "Knowledge Based Activity Recognition with Dynamic Bayesian Network," in *Computer Vision -- ECCV 2010*, 2010, pp. 532–546.

[32] M. Ziaeefard and R. Bergevin, "Semantic human activity recognition: A literature review," *Pattern Recognit.*, vol. 48, no. 8, pp. 2329–2345, 2015, doi: https://doi.org/10.1016/j.patcog.2015.03.006.

[33] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and F.-F. Li, "Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos," *CoRR*, vol. abs/1507.05738, 2015, [Online]. Available: http://arxiv.org/abs/1507.05738.

[34] L. Wang, Y. Qiao, and X. Tang, "Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors," *CoRR*, vol. abs/1505.04868, 2015, [Online]. Available: http://arxiv.org/abs/1505.04868.

[35] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, 1986, doi: 10.1007/BF00116251.

[36] B. W. Silverman and M. C. Jones, "E. Fix and J.L. Hodges (1951): An Important

Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951),” *Int. Stat. Rev. / Rev. Int. Stat.*, vol. 57, no. 3, pp. 233–238, May 1989, doi: 10.2307/1403796.

[37] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995, vol. 1, pp. 278–282 vol.1, doi: 10.1109/ICDAR.1995.598994.

[38] A. Liaw and M. Wiener, “Classification and Regression by RandomForest,” *Forest*, vol. 23, 2001.

[39] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943, doi: 10.1007/BF02478259.

[40] C. Cortes and V. Vapnik, “Support-vector networks,” *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[41] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, “Support Vector Clustering,” *J. Mach. Learn. Res.*, vol. 2, pp. 125–137, Mar. 2002.

[42] Y. Kong, Y. Jia, and Y. Fu, “Learning Human Interaction by Interactive Phrases,” in *Computer Vision -- ECCV 2012*, 2012, pp. 300–313.

[43] C. Schuldt, I. Laptev, and B. Caputo, “Recognizing human actions: a local SVM approach,” in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, 2004, vol. 3, pp. 32-36 Vol.3, doi: 10.1109/ICPR.2004.1334462.

[44] S. G. Bhele, V. H. Mankar, and others, “A review paper on face recognition techniques,” *Int. J. Adv. Res. Comput. Eng. \& Technol.*, vol. 1, no. 8, pp. 339–

346, 2012.

[45] M. N. A. H. Sha’abani, N. Fuad, N. Jamal, and M. F. Ismail, “kNN and SVM Classification for EEG: A Review BT - InECCE2019,” 2020, pp. 555–565.

[46] W. Pitts, “Some observations on the simple neuron circuit,” *Bull. Math. Biophys.*, vol. 4, no. 3, pp. 121–129, 1942, doi: 10.1007/BF02477942.

[47] H. J. KELLEY, “Gradient Theory of Optimal Flight Paths,” *ARS J.*, vol. 30, no. 10, pp. 947–954, 1960, doi: 10.2514/8.5282.

[48] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980, doi: 10.1007/BF00344251.

[49] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[50] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[51] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[52] C. Szegedy *et al.*, “Going Deeper with Convolutions,” *CoRR*, vol. abs/1409.4842, 2014, [Online]. Available: http://arxiv.org/abs/1409.4842.

[53] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.

[54] A. Krizhevsky, I. Sutskever, and G. Hinton, “ImageNet Classification with Deep

Convolutional Neural Networks," *Neural Inf. Process. Syst.*, vol. 25, 2012, doi: 10.1145/3065386.

[55] J. Zegers and H. Van hamme, "{CNN-LSTM} models for Multi-Speaker Source Separation using Bayesian Hyper Parameter Optimization," *CoRR*, vol. abs/1912.09254, 2019, [Online]. Available: http://arxiv.org/abs/1912.09254.

[56] C. Etienne, G. Fidanza, A. Petrovskii, L. Devillers, and B. Schmauch, "Speech Emotion Recognition with Data Augmentation and Layer-wise Learning Rate Adjustment," *CoRR*, vol. abs/1802.05630, 2018, [Online]. Available: http://arxiv.org/abs/1802.05630.

[57] T. Akilan, Q. J. Wu, A. Safaei, J. Huo, and Y. Yang, "A 3D CNN-LSTM-based image-to-image foreground segmentation," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 3, pp. 959–971, 2019.

[58] L. Wang and D. Suter, "Recognizing Human Activities from Silhouettes: Motion Subspace and Factorial Discriminative Graphical Model," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8, doi: 10.1109/CVPR.2007.383298.

[59] S. Chun and C.-S. Lee, "Human Action Recognition Using Histogram of Motion Intensity and Direction from Multi View," *IET Comput. Vis.*, vol. 10, 2016, doi: 10.1049/iet-cvi.2015.0233.

[60] Q. Xiong, J. Zhang, P. Wang, D. Liu, and R. X. Gao, "Transferable two-stream convolutional neural network for human action recognition," *J. Manuf. Syst.*, vol. 56, pp. 605–614, 2020.

[61] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action

recognition in videos," *arXiv Prepr. arXiv1406.2199*, 2014.

[62] A. S. Cakmak *et al.*, "Late fusion of machine learning models using passively captured interpersonal social interactions and motion from smartphones predicts decompensation in heart failure." 2021.

[63] N. Kapinski *et al.*, "Late fusion of deep learning and hand-crafted features for Achilles tendon healing monitoring." 2019.

[64] A. Walsman *et al.*, "Early Fusion for Goal Directed Robotic Vision," *CoRR*, vol. abs/1811.08824, 2018, [Online]. Available: http://arxiv.org/abs/1811.08824.

[65] P. Saleiro, N. Milic-Frayling, E. M. Rodrigues, and C. Soares, "Early Fusion Strategy for Entity-Relationship Retrieval," *CoRR*, vol. abs/1707.09075, 2017, [Online]. Available: http://arxiv.org/abs/1707.09075.

[66] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional Two-Stream Network Fusion for Video Action Recognition," *CoRR*, vol. abs/1604.06573, 2016, [Online]. Available: http://arxiv.org/abs/1604.06573.

[67] K. Gadzicki, R. Khamsehashari, and C. Zetzsche, "Early vs Late Fusion in Multimodal Convolutional Neural Networks," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, 2020, pp. 1–6, doi: 10.23919/FUSION45008.2020.9190246.

[68] K. Liu, Y. Li, N. Xu, and P. Natarajan, "Learn to Combine Modalities in Multimodal Deep Learning." 2018.

[69] J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? {A} New Model and the Kinetics Dataset," *CoRR*, vol. abs/1705.07750, 2017, [Online]. Available: http://arxiv.org/abs/1705.07750.

[70] M. Vrigkas, C. Nikou, and I. A. Kakadiaris, “A Review of Human Activity Recognition Methods,” *Front. Robot. AI*, vol. 2, p. 28, 2015, doi: 10.3389/frobt.2015.00028.

[71] C. A. Ronao and S.-B. Cho, “Human activity recognition with smartphone sensors using deep learning neural networks,” *Expert Syst. Appl.*, vol. 59, pp. 235–244, 2016, doi: https://doi.org/10.1016/j.eswa.2016.04.032.

[72] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, “Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition,” in *Proceedings of the 24th International Conference on Artificial Intelligence*, 2015, pp. 3995–4001, [Online]. Available: http://dl.acm.org/citation.cfm?id=2832747.2832806.

[73] P. Vepakomma, D. De, S. Das, and S. Bhansali, “A-Wristocracy: Deep Learning on Wrist-worn Sensing for Recognition of User Complex Activities,” 2015, doi: 10.1109/BSN.2015.7299406.

[74] X. Li, Y. Zhang, M. Li, I. Marsic, J. Yang, and R. S. Burd, “Deep Neural Network for RFID-Based Activity Recognition,” *Proc. Eighth Wirel. Students, by Students, Students Work. Work. Wirel. Students, by Students, Students (8th 2016 New York, N.Y.)*, vol. 2016, pp. 24–26, Oct. 2016, doi: 10.1145/2987354.2987355.

[75] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, “A Public Domain Dataset for Human Activity Recognition using Smartphones,” 2013.

[76] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, “Activity recognition using cell phone accelerometers,” *ACM SIGKDD Explor. Newsl.*, vol. 12, pp. 74–82, 2011.

[77] R. Chavarriaga *et al.*, “The Opportunity challenge: A benchmark database for on-

body sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, 2013, doi: https://doi.org/10.1016/j.patrec.2012.12.014.

[78] D. Roggen *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, 2010, pp. 233–240, doi: 10.1109/INSS.2010.5573462.

[79] S. S. Saha, S. Rahman, M. J. Rasna, A. K. M. Mahfuzul Islam, and M. A. Rahman Ahad, "DU-MD: An Open-Source Human Action Dataset for Ubiquitous Wearable Sensors," in *2018 Joint 7th International Conference on Informatics, Electronics Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision Pattern Recognition (icIVPR)*, 2018, pp. 567–572, doi: 10.1109/ICIEV.2018.8641051.

[80] T. J. McCue *et al.*, "Digital Farming decisions and insights to maximize every acre," *IEEE Trans. Ind. Informatics*, vol. 14, no. 9, pp. 192–202, Nov. 2018, doi: https://doi.org/10.1016/j.comnet.2016.01.009.

[81] M. Zhang and A. A. Sawchuk, "USC-HAD: A Daily Activity Dataset for Ubiquitous Activity Recognition Using Wearable Sensors," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1036–1043, doi: 10.1145/2370216.2370438.

[82] G. Bhat, N. Tran, H. Shill, and U. Y. Ogras, "w-HAR: An Activity Recognition Dataset and Framework Using Low-Power Wearable Devices," *Sensors (Basel).*, vol. 20, no. 18, p. 5356, Sep. 2020, doi: 10.3390/s20185356.

[83] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of Smartphone Motion Sensors for Physical Activity Recognition," *Sensors*, vol. 14,

no. 6, pp. 10146–10176, 2014, [Online]. Available: http://www.mdpi.com/1424-8220/14/6/10146.

[84] D. Micucci, M. Mobilio, and P. Napoletano, “UniMiB {SHAR:} a new dataset for human activity recognition using acceleration data from smartphones,” *CoRR*, vol. abs/1611.0, 2016, [Online]. Available: http://arxiv.org/abs/1611.07688.

[85] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, “Actions as Space-Time Shapes,” in *The Tenth IEEE International Conference on Computer Vision (ICCV’05)*, 2005, pp. 1395–1402.

[86] M. D. Rodriguez, J. Ahmed, and M. Shah, “Action MACH a spatio-temporal Maximum Average Correlation Height filter for action recognition,” 2008, doi: 10.1109/CVPR.2008.4587727.

[87] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8, doi: 10.1109/CVPR.2008.4587756.

[88] M. Marszalek, I. Laptev, and C. Schmid, “Actions in context,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2929–2936, doi: 10.1109/CVPR.2009.5206557.

[89] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB51: A Large Video Database for Human Motion Recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 2556–2563, doi: 10.1109/ICCV.2011.6126543.

[90] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale Video Classification with Convolutional Neural Networks,” 2014.

[91] W. Kay *et al.*, “The Kinetics Human Action Video Dataset,” *CoRR*, vol. abs/1705.0, 2017, [Online]. Available: http://arxiv.org/abs/1705.06950.

[92] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, and A. Zisserman, “A Short Note about Kinetics-600,” *CoRR*, vol. abs/1808.0, 2018, [Online]. Available: http://arxiv.org/abs/1808.01340.

[93] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, “A Short Note on the Kinetics-700 Human Action Dataset,” *CoRR*, vol. abs/1907.06987, 2019, [Online]. Available: http://arxiv.org/abs/1907.06987.

[94] Z. Sun, J. Liu, Q. Ke, H. Rahmani, M. Bennamoun, and G. Wang, “Human Action Recognition from Various Data Modalities: {A} Review,” *CoRR*, vol. abs/2012.11866, 2020, [Online]. Available: https://arxiv.org/abs/2012.11866.

[95] K. Soomro, A. R. Zamir, and M. Shah, “{UCF101:} {A} Dataset of 101 Human Actions Classes From Videos in The Wild,” *CoRR*, vol. abs/1212.0, 2012, [Online]. Available: http://arxiv.org/abs/1212.0402.

[96] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep High-Resolution Representation Learning for Human Pose Estimation,” *CoRR*, vol. abs/1902.09212, 2019, [Online]. Available: http://arxiv.org/abs/1902.09212.

[97] J. Shotton *et al.*, “Real-time human pose recognition in parts from single depth images,” in *CVPR 2011*, 2011, pp. 1297–1304, doi: 10.1109/CVPR.2011.5995316.

[98] J. Sung, C. Ponce, B. Selman, and A. Saxena, “Human activity detection from RGBD images,” 2011.

[99] H. S. Koppula, R. Gupta, and A. Saxena, “Learning Human Activities and Object Affordances from {RGB-D} Videos,” *CoRR*, vol. abs/1210.1, 2012, [Online].

Available: http://arxiv.org/abs/1210.1207.

[100] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, “NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis,” Jun. 2016.

[101] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, “{NTU} {RGB+D} 120: {A} Large-Scale Benchmark for 3D Human Activity Understanding,” *CoRR*, vol. abs/1905.04757, 2019, [Online]. Available: http://arxiv.org/abs/1905.04757.

[102] F. Liu, C. Shen, and G. Lin, “Deep Convolutional Neural Fields for Depth Estimation from a Single Image,” *CoRR*, vol. abs/1411.6387, 2014, [Online]. Available: http://arxiv.org/abs/1411.6387.

[103] D. Eigen and R. Fergus, “Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture,” *CoRR*, vol. abs/1411.4734, 2014, [Online]. Available: http://arxiv.org/abs/1411.4734.

[104] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 1290–1297, doi: 10.1109/CVPR.2012.6247813.

[105] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, “Cross-view Action Modeling, Learning and Recognition,” *CoRR*, vol. abs/1405.2941, 2014, [Online]. Available: http://arxiv.org/abs/1405.2941.

[106] H. Rahmani, A. Mahmood, D. Q Huynh, and A. Mian, “HOPC: Histogram of Oriented Principal Components of 3D Pointclouds for Action Recognition,” in *Computer Vision -- ECCV 2014*, 2014, pp. 742–757.

[107] C. Gao *et al.*, “InfAR dataset: Infrared action recognition at different times,” *Neurocomputing*, vol. 212, pp. 36–47, 2016, doi: 10.1016/j.neucom.2016.05.094.

[108] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, “Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification,” in *Computer Vision -- ECCV 2010*, 2010, pp. 392–405.

[109] B. G. Fabian Caba Heilbron Victor Escorcia and J. C. Niebles, “ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[110] S. Abu-El-Haija *et al.*, “YouTube-8M: A Large-Scale Video Classification Benchmark,” 2016, [Online]. Available: https://arxiv.org/pdf/1609.08675v1.pdf.

[111] R. Goyal *et al.*, “The ‘something something’ video database for learning and evaluating visual common sense,” *CoRR*, vol. abs/1706.0, 2017, [Online]. Available: http://arxiv.org/abs/1706.04261.

[112] C. Gu *et al.*, “{AVA:} {A} Video Dataset of Spatio-temporally Localized Atomic Visual Actions,” *CoRR*, vol. abs/1705.08421, 2017, [Online]. Available: http://arxiv.org/abs/1705.08421.

[113] F. Wang, J. Feng, Y. Zhao, X. Zhang, S. Zhang, and J. Han, “Joint Activity Recognition and Indoor Localization With WiFi Fingerprints,” *IEEE Access*, vol. 7, p. 1, 2019, doi: 10.1109/ACCESS.2019.2923743.

[114] I. Cleland *et al.*, “Optimal Placement of Accelerometers for the Detection of Everyday Activities,” *Sensors*, vol. 13, no. 7, pp. 9183–9200, 2013, [Online]. Available: http://www.mdpi.com/1424-8220/13/7/9183.

[115] H. Gjoreski, J. Bizjak, M. Gjoreski, and M.Gams, “Comparing deep a classical machine learning methods for human activity recognition using wrist accelerometer,” in *Proceedings of the 25th International Conference on Artificial Intelligence*, 2016, pp. 1–7.

[116] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. T. Pulickal, “Design Considerations for the WISDM Smart Phone-based Sensor Mining Architecture,” in *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data*, 2011, pp. 25–33, doi: 10.1145/2003653.2003656.

[117] S. M. Safdarnejad, X. Liu, L. Udpa, B. Andrus, J. Wood, and D. Craven, “Sports Videos in the Wild (SVW): A Video Dataset for Sports Analysis,” May 2015.

[118] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “{C3D:} Generic Features for Video Analysis,” *CoRR*, vol. abs/1412.0, 2014, [Online]. Available: http://arxiv.org/abs/1412.0767.

[119] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks.” 2015.

[120] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.

[121] H. Pham, Q. Xie, Z. Dai, and Q. V Le, “Meta Pseudo Labels,” *CoRR*, vol. abs/2003.10580, 2020, [Online]. Available: https://arxiv.org/abs/2003.10580.

[122] S. J. Pan and Q. Yang, “A Survey on Transfer Learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010, doi: 10.1109/TKDE.2009.191.

[123] F. Chen, D. Delannay, and C. De Vleeschouwer, “An autonomous framework to produce and distribute personalized team-sport video summaries: A basketball case study,” *IEEE Trans. Multimed.*, vol. 13, no. 6, pp. 1381–1394, 2011.

[124] K. Avetisyan and T. Ghukasyan, “Word Embeddings for the Armenian Language: Intrinsic and Extrinsic Evaluation,” *CoRR*, vol. abs/1906.03134, 2019, [Online]. Available: http://arxiv.org/abs/1906.03134.

[125] Y. Wang, S. Inguva, and B. Adsumilli, “YouTube {UGC} Dataset for Video Compression Research,” *CoRR*, vol. abs/1904.06457, 2019, [Online]. Available: http://arxiv.org/abs/1904.06457.

[126] X. Zhang, J. J. Zhao, and Y. LeCun, “Character-level Convolutional Networks for Text Classification,” *CoRR*, vol. abs/1509.01626, 2015, [Online]. Available: http://arxiv.org/abs/1509.01626.

[127] Yang Wang, Hao Jiang, M. S. Drew, Ze-Nian Li, and G. Mori, “Unsupervised Discovery of Action Classes,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, Jun. 2006, vol. 2, pp. 1654–1661, doi: 10.1109/CVPR.2006.321.

[128] W. J. McNally, K. Vats, T. Pinto, C. Dulhanty, J. McPhee, and A. Wong, “GolfDB: {A} Video Database for Golf Swing Sequencing,” *CoRR*, vol. abs/1903.06528, 2019, [Online]. Available: http://arxiv.org/abs/1903.06528.

[129] V. Kazemi, M. Burenius, H. Azizpour, and J. Sullivan, “Multi-view body part recognition with random forests,” 2013, doi: 10.5244/C.27.48.

[130] K. Rangasamy, M. A. As’ari, N. A. Rahmad, and N. F. Ghazali, “Hockey activity recognition using pre-trained deep learning model,” *ICT Express*, vol. 6, no. 3, pp.

170–174, 2020, doi: https://doi.org/10.1016/j.icte.2020.04.013.

[131] S. Su, J. Pyo Hong, J. Shi, and H. Soo Park, "Predicting Behaviors of Basketball Players From First Person Videos," Jul. 2017.

[132] Q. H. Lam, Q.-D. Le, K. Van Nguyen, and N. L.-T. Nguyen, "UIT-ViIC: {A} Dataset for the First Evaluation on Vietnamese Image Captioning," *CoRR*, vol. abs/2002.00175, 2020, [Online]. Available: https://arxiv.org/abs/2002.00175.

[133] G.-G. Lee, H. Kim, and W.-Y. Kim, "Highlight generation for basketball video using probabilistic excitement," in *2009 IEEE International Conference on Multimedia and Expo*, 2009, pp. 318–321, doi: 10.1109/ICME.2009.5202499.

[134] R. Shah and R. Romijnders, "Applying Deep Learning to Basketball Trajectories," *CoRR*, vol. abs/1608.03793, 2016, [Online]. Available: http://arxiv.org/abs/1608.03793.

[135] K. Wang and R. Zemel, "Classifying NBA Offensive Plays Using Neural Networks," 2016.

[136] N. Liu and J. Han, "A deep spatial contextual long-term recurrent convolutional network for saliency detection," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3264–3274, 2018.

[137] M. Abdullah, M. Ahmad, and D. Han, "Facial Expression Recognition in Videos: An CNN-LSTM based Model for Video Classification," in *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, 2020, pp. 1–3.

[138] J. You and J. Korhonen, "Attention Boosted Deep Networks For Video Classification," in *2020 IEEE International Conference on Image Processing*

(ICIP), 2020, pp. 1761–1765.

[139] X. Peng and C. Schmid, “Multi-region two-stream R-CNN for action detection,” in *European conference on computer vision*, 2016, pp. 744–759.

[140] R. Hetherington, “The Perception of the Visual World. By James J. Gibson. USA: Houghton Mifflin Company, 1950 (George Allen & Unwin, Ltd., London). Price 35s.,” *J. Ment. Sci.*, vol. 98, no. 413, p. 717, 1952.

[141] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in *Scandinavian conference on Image analysis*, 2003, pp. 363–370.

[142] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, 2012.

[143] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, “Convolutional learning of spatio-temporal features,” in *European conference on computer vision*, 2010, pp. 140–153.

[144] K. Xu, Z. Qin, and G. Wang, “Recognize human activities from multi-part missing videos,” in *2016 IEEE International Conference on Multimedia and Expo (ICME)*, 2016, pp. 1–6, doi: 10.1109/ICME.2016.7552941.

[145] F. Negin, M. Koperski, C. F. Crispim, F. Bremond, S. Coşar, and K. Avgerinakis, “A hybrid framework for online recognition of activities of daily living in real-world settings,” in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2016, pp. 37–43, doi: 10.1109/AVSS.2016.7738021.

[146] S. S. Rautaray and A. Agrawal, “Vision Based Hand Gesture Recognition for

Human Computer Interaction: A Survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2015, doi: 10.1007/s10462-012-9356-9.

[147] M. Rohrbach *et al.*, "Recognizing Fine-Grained and Composite Activities using Hand-Centric Features and Script Data," *CoRR*, vol. abs/1502.06648, 2015, [Online]. Available: http://arxiv.org/abs/1502.06648.

[148] L. Guo, L. Wang, J. Liu, W. Zhou, and B. Lu, "HuAc: Human Activity Recognition Using Crowdsourced WiFi Signals and Skeleton Data," *Wirel. Commun. Mob. Comput.*, vol. 2018, p. 6163475, 2018, doi: 10.1155/2018/6163475.

[149] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding," *CoRR*, vol. abs/1604.01753, 2016, [Online]. Available: http://arxiv.org/abs/1604.01753.

[150] K. Akila and S. Chitrakala, "Highly refined human action recognition model to handle intraclass variability & interclass similarity," *Multimed. Tools Appl.*, vol. 78, no. 15, pp. 20877–20894, 2019, doi: 10.1007/s11042-019-7392-z.

[151] A. Jalal, Y.-H. Kim, Y.-J. Kim, S. Kamal, and D. Kim, "Robust Human Activity Recognition from Depth Video Using Spatiotemporal Multi-Fused Features," *Pattern Recognit.*, vol. 61, 2016, doi: 10.1016/j.patcog.2016.08.003.

[152] L. Zheng *et al.*, "A Novel Energy-Efficient Approach for Human Activity Recognition.," *Sensors (Basel).*, vol. 17, no. 9, Sep. 2017, doi: 10.3390/s17092064.

[153] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "InnoHAR: A Deep Neural Network for Complex Human Activity Recognition," *IEEE Access*, vol. 7, pp.

9893–9902, 2019, doi: 10.1109/ACCESS.2018.2890675.

.