A HYBRID MODEL FOR HUMAN ACTIVITY SURVEILLANCE USING DEEP LEARNING AND MOTION FLOW

A Thesis Submitted

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY IN

Computer Science & Engineering

By

Palak Girdhar

Registration No. – 17SCSE301002

Supervisor: Dr. Prashant Johri, SCSE, Galgotias University, Greater Noida

Co Supervisor: Dr. Deepali Virmani, CSE Deptt,

Vivekananda Institute of Professional Studies-Technical Campus, Delhi

GALGOTIAS UNIVERSITY UTTAR PRADESH

March-2022

Galgotias University
Uttar Pradesh
School of Computing Science and Engineering

CERTIFICATE

This is to certify that Ms. Palak Girdhar has presented her pre-submission seminar of the thesis entitled "**A Hybrid Model for Human Activity Surveillance using Deep Learning and Motion Flow**" before the committee and summary is approved and forwarded to School Research Committee of School of Computing Science and Engineering, in the Faculty of Engineering & Technology, Galgotias University, Uttar Pradesh.

Dr. Prashant Johri
Supervisor & Professor, SCSE
Galgotias University, Uttar Pradesh

Dr. Deepali Virmani
Co-Supervisor & Professor, CSE
VIPS-TC, New Delhi

Dean
School of Computing Science and Engineering
Galgotias University, Uttar Pradesh

Dean PG & PhD
Galgotias University
Uttar Pradesh

Date:

APPROVAL SHEET

This thesis entitled **"A Hybrid Model for Human Activity Surveillance using Deep Learning and Motion Flow"** by **Palak Girdhar** is approved for the degree of DOCTOR OF PHILOSOPHY in Computer Science and Engineering.

Examiners

____

____

Supervisor (s)

____

____

____

Chairman

____

Date:

Place:

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis, entitled "**A Hybrid Model for Human Activity Surveillance using Deep Learning and Motion Flow**" in fulfilment of the requirements for the award of the degree of Doctor of Philosophy in Computer Science and Engineering, submitted in Galgotias University, Greater Noida is an authentic record of my own work carried out during a period from **September 2017 – December 2021** under the supervision of **Dr. Prashant Johri** and co-supervision of **Dr. Deepali+ Virmani**.

The matter embodied in this thesis has not been submitted by me for the award of any other degree of this or any other University/Institute.

Palak Girdhar

This is to certify that the above statement made by the candidate is correct to the best of our knowledge.

Dr.Prashant Johri	**Dr. Deepali Virmani**
Supervisor	Co-Supervisor
SCSE, Galgotias University	Deptt. of Computer Science, VIPS-TC

The Ph.D. Viva-Voice examination of Ms. Palak Girdhar, Research Scholar, has been held on__________.

Sign. of Supervisor(s)   Sign. of Co-Supervisor(s)   Sign. of External Examiner

ACKNOWLEDGMENT

I thank God for everything that has been bestowed on me. I thank God for all the experiences and moments that have made me a better person. The completion of this thesis would not have been possible without God’s blessings. The timely completion of my work involves my association with many people. I would like to thank all those who made it possible.

First and foremost, I would like to express my sincere gratitude to my supervisor, Dr. Prashant Johri, Professor, SCSE, Galgotias University, Greater Noida. He is a gem of a person. His positive attitude towards life always gave me strength to move ahead. I am thankful to him for his constant support.

My heartfelt thanks to my co-supervisor, Dr. Deepali Virmani, Professor, Vivekanand Institute of Professional Studies-Technical Campus. She is a very charming and wonderful person. She motivated me throughout the time period.

I owe my deepest gratitude to my husband, Madhur Girdhar, for his eternal support and never dying patience which has helped me in completing this work. His infallible love and affection have been my constant strength. I would like to deeply thank, my loving children, Advik and Ayaan for their endless love and patience. My special thanks to my in-laws, my parents, brother, sister and friends for believing and supporting me.

My deepest thanks to my friend Dr. Charu Gupta, Assistant Professor, Bhagwan Parshuram Institute of Technology, Delhi for her endless support to accomplish this task. She has been the pillar of my strength. Whenever I felt low, she encouraged me to move further. She is the guiding force behind my all endeavours. Without her support, it won’t be possible for me to reach this milestone.

I would like to thank my tutor Mr. Ravi Kant Tyagi, for his valuable guidance throughout my journey. He provided me with the tools that I needed to choose the right direction and successfully completion of my work.

I am also grateful to Prof. Payal Pahwa, Principal, Bhagwan Parshuram Institute of Technology, Delhi for her encouragement and constant support at my workplace. Her love and blessings have always been fruitful for me. I thank her for giving me a cordial environment for carrying out my research.

Finally, I could not have completed this work without the support of my friends, Ms. Tanu Sharma and Ms. Anamika Bhatia who motivated and provided happy distractions to rest my mind outside of my research.

Palak Girdhar

ABSTRACT

Human Activity Recognition from digital video is gaining popularity because of increasing crime rate worldwide. With the increase in crime rate, there is a massive increase in the installation of surveillance cameras. Monitoring these surveillance cameras on 24x7 basis, requires a lot of manpower. The accuracy of the system depends upon the activeness of the supervising entity. Sometimes, biased nature of the supervising person, fatigue issue results in the inaccurate decision. Therefore, there is a dire need of automated surveillance system, which can detect anomalous event correctly and in time. The problem of detecting anomalous event involves recognizing the human activity closely from the data collected through CCTVs.

For the effective monitoring, an automated surveillance system is the need of time. In this work, a deep learning-based approach is used to classify the activities correctly from the video data. Collection of Image frames are considered for classification, as action in a single image frame may not be same in subsequent image frame. Therefore, in order to understand the human activities, analysis is made on sequence of image frames. In addition to spatial correlation present in the 2D images, temporal structure present in the video data is also considered.

To classify human activities from the video data, a method based on Deep Learning (DL) is proposed. Recently, Convolutional Neural Network (CNN) has shown great improvement in the field of computer vision with the application of image and video classification. Advancement in the architectures of CNN is explored and analysed. Based on structural characteristics of explored CNNs and considering the accuracy achieved by various architectures in ILSVRC (Image Large Scale Visual Recognition Challenge), a method is proposed for the classification of human activities. In addition to spatial correlation present in 2D images, video data also own the corelation in temporal domain. A method based on Inception and LSTM, *Incept_LSTM,* is proposed. The proposed method is able to recognize human activities efficiently. Further, importance of hyper-parameter tuning is explored and implemented. The proposed method is trained and validated on UCF-Crime dataset. Experimental results validate that the proposed method has effectively identified human actions in videos.

Keywords: Human Action Recognition, Convolutional Neural Network, Deep Learning, Inception-v3, LSTM, Hyper-parameter tuning.

.TABLE OF CONTENTS

Title ***Page No.***

Candidate's Declaration ***ii***
Acknowledgment ***iii***
Abstract ***v***
List of Figures ***vi***
List of Tables ***vii***
List of Publications ***ix***
Abbreviations ***x***

Chapter 1: Introduction **1-26**

1.1 Computer Vision 1
1.2 Goal of Computer Vision 1
1.3 Computer Vision: Major Applications Areas 2
1.4 Computer Vision: For Security and Safety applications 4
1.4.1 Human Activity Recognition: Behavioural Perspective 7
1.4.2 Human Activity Recognition: Surveillance as an Application 8
1.4.3 Challenges 10
1.5 Machine Learning: A Deep Learning Perspective 10
1.5.1 Need and Advantage 11
1.6 Deep Learning Models: Activity Recognition 12
1.6.1 CNN 13
1.6.2 RNN 14
1.7 Building Blocks of Deep Learning Models 14
1.7.1 Activation Function 14
1.7.2 Loss Function 18
1.7.3 Role of Optimizers and types 19
1.7.4 Hyperparameters 24
1.8 Objective 25
1.9 Block Diagram of the Proposed Work 26
1.10 Organization of Thesis 26
1.11 Summary 26

Title			*Page No.*
Chapter 2: Literature Review			**27-36**
2.1	Introduction		27
	2.1.1	Human activity recognition using deep learning	27
	2.1.2	Types of HAR	29
2.2	Hybrid Inception and LSTM for Behavioural Analysis		30
	2.2.1	LSTM for behavioral Analysis	32
	2.2.2	Inception based models for behavioural Analysis	34
2.3	Summary		36
Chapter 3: Evolution of Deep Learning for Image Classification			**37-55**
3.1	Convolutional Neural Network (CNN)		37
3.2	Major Challenges		40
3.3	CNN Architectures for Image Classification		45
	3.3.1	Pre-Trained Models	45
3.4	Evaluation Criteria for CNN Models		53
3.5	Analysis of various CNN Architectures		54
3.6	Summary		55
Chapter 4: Inception and LSTM based Methodology for Human Activity Recognition			**56-69**
4.1	Background		56
4.2	Motivation		59
4.3	Proposed Architecture- Incept_LSTM		59
	4.3.1	Spatial Feature Extraction	60
	4.3.2	Temporal Feature Extraction	61
4.4	Experimental Settings and Initial Preparation		62
4.5	Performance Measure		64
4.6	Experimental Results and Analysis		66
4.7	Summary		69

Title *Page No.*

Chapter 5: A Human Activity Recognition Model based on Motion Flow using Inception and LSTM 70-82

5.1 Introduction 70

5.2 Motivation 70

5.3 Proposed motion flow based Incept_LSTM architecture 71

5.4 Experimental Results and Analysis 74

5.4.1 Settings in Training and Testing Phase 74

5.4.2 Analysis 76

5.5 Hyper-Parameters' impact on Model Performance 76

5.6 Comparison with state-of-the-art methods 81

5.7 Summary 82

Chapter 6: Conclusion and Future Scope 83-86

6.1 Conclusion 83

6.2 Future Scope 86

References 87-96

LIST OF FIGURES

Figures			*Page No.*
Figure 1.1	**:**	Asia Pacific Video Surveillance Market Size, 2018(USD Billions) (Fortune Business Insights)	5
Figure 1.2	**:**	Video Surveillance Market Share	6
Figure 1.3	**:**	Levels of Human Activities	9
Figure 1.4	**:**	Sigmoid Curve	16
Figure 1.5	**:**	Tanh Curve	16
Figure 1.6	**:**	ReLU Curve	17
Figure 1.7	**:**	SGD without Momentum (left) SGD with momentum (right)	21
Figure 1.8	**:**	Block Diagram of the Proposed Work	26
Figure 3.1	**:**	CNN Architecture	38
Figure 3.2	**:**	A Convolutional Operation	39
Figure 3.3	**:**	Max pooling Operation	40
Figure 3.4	**:**	Centre cropping (Left) and random cropping (right)	42
Figure 3.5	**:**	Random Images showing horizontally (left) and vertical (right) shift.	42
Figure 3.6	**:**	The Sigmoid Function and its derivative	44
Figure 3.7	**:**	LeNet Architecture	46
Figure 3.8	**:**	AlexNet Architecture	47
Figure 3.9	**:**	Architecture of ZfNet	47
Figure 3.10	**:**	Architecture of VGGNet (configuration D, VGG16)	48
Figure3.11	**:**	Graphical representation of factors into smaller convolutions	49
Figure 3.12	**:**	Graphical representation of Factorization into Asymmetric Convolutions	50
Figure 3.13	**:**	Inception-v3 Network Architecture	51
Figure 3.14	**:**	Inception Module A	51

Figures *Page No.*

Figure 3.15 : Inception Module B 52

Figure 3.16 : Inception Module C 52

Figure 3.17 : Residual block 53

Figure 4.1 : A LSTM Memory Cell 57

Figure 4.2 : Proposed Approach Incept_LSTM 62

Figure 4.3 : UCF-Crime Dataset Categories 63

Figure 4.4 : Confusion matrix of action recognition using Incept_LSTM 68

Figure 4.5 : Model Accuracy and Loss 68

Figure 5.1 : Schematic Representation of Motion flow based Incept_LSTM 73

Figure 5.2 : Optical Flow Computation Using Lucas Kanade Method 74

Figure 5.3 : Accuracy Comparison of Various Optimizers 77

Figure 5.4 : Loss Comparison of Various Optimizers 77

Figure 5.5 : Accuracy Comparison of Various Activation Function 78

Figure 5.6 : Loss Comparison of Various Activation Function 79

Figure 5.7 : Accuracy Comparison on Various Learning Rates 80

Figure 5.8 : Loss Comparison on Various Learning Rates 80

LIST OF TABLES

Tables		*Page No.*
Table 3.1 :	CNNs Architecture's analysis	54
Table 4.1 :	Incept_LSTM Results	66
Table 4.2 :	Accuracy comparison of the proposed method with existing methods	67
Table 4.3 :	F1-Score Comparison of Incept_LSTM with existing work	67
Table 5.1 :	Performance Comparison on Various Optimizers	77
Table 5.2 :	Performance Comparison on Various Activation Function	78
Table 5.3 :	Performance Comparison on Various Learning Rate	80
Table 5.4 :	Motion flow based Incept_LSTM Results	81
Table 5.5 :	Accuracy comparison with State-of-the-art methods	81
Table 5.6 :	Comparison of the proposed method with state-of-the-art in terms of the parameters and time complexity	82

LIST OF PUBLICATIONS

1) Palak Girdhar, S. Sarvana Kumar, Deepali Virmani, “A hybrid fuzzy framework for Face Detection and Recognition using Behavioral Traits” published in “Journal of Statistics and Management Systems”, Vol.22, Issue :2, 2019, DOI:10.1080/09720510.2019.1580905, PP: 271-281. **(ESCI)**

2) Palak Girdhar, Prashant Johri, Deepali Virmani, “Incept_LSTM: Accession for human activity concession in automatic surveillance” published in “Journal of Discrete Mathematical Sciences and Cryptography”, Vol.24, Issue:4, 2020, DOI: 10.1080/09720529 .2020.1804132, PP: 1-15. **(SCOPUS)**

3) Palak Girdhar, Prashant Johri, Deepali Virmani, “Vision Based Human Activity Recognition: A Comprehensive Review of Methods & Techniques” published in Turkish Journal of Computer and Mathematics”, Vol. 12, Issue 10, 2021, PP: 7383-7394. **(SCOPUS)**

4) Palak Girdhar, Prashant Johri, Deepali Virmani, “An Optimized Incept_LSTM for Human Activity Recognition”, (*Accepted for Publication)* in “Journal of Interdisciplinary Mathematics”, (**SCOPUS)**

5) Palak Girdhar, Prashant Johri, Deepali Virmani, “Evolution of Deep Learning for Image Classification: Methods, Challenges and Architectures” in ICADCML conference, NIT Warangal. **(SCOPUS Conference)**

ABBREVIATIONS

Abbreviation		**Description**
CNN	:	Convolutional Neural Networks
HAR	:	Human Activity Recognition
AI	:	Artificial Intelligence
DL	:	Deep Learning
ML	:	Machine Learning
EIN	:	Equivalent Inception Networks
TL	:	Transfer Learning
WSN	:	Wireless Sensor Networks
ROC	:	Receiver-Operating Characteristic Curve
WLSTM	:	Weighted Long Short-Term Memory Network
RNN	:	Recurrent Neural Networks
LSTM	:	Long Short-Term Memory
IRCNN	:	Inception Recurrent Convolutional Neural Network
GAP	:	Global Average Pooling Layer
BN	:	Batch Normalization Layer

CHAPTER 1

INTRODUCTION

1.1 Computer Vision

Computer vision (Jahne, 2000) is a field of Artificial Intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information. It is emerged with dual goal. Firstly, from the biological science perspective, it aims to develop computational models of human visual cortex (Schmidhuber, 2015) system and other engineering perspective aims to develop autonomous system that could perform better than human visual system. The mentioned goals are interrelated in most manner. The characteristics of visual system gives inspiration for the development of models that mimic human visual system.

Larry Roberts, who is known to be father of Computer Vision (Shapiro, 2020), also the father of ARPANET in his Ph.D thesis at MIT, had worked on 3D geometrical transformation from 2D input data. Many researchers at MIT followed his work and studied computer vision in the context of blocks. Later on, many researchers worked on low level features of the image like edge detection and segmentation. In 1978, a major milestone was achieved by David Marr who took a bottom-up approach for scene understanding (Rosenfeld, 1997).

1.2 Goal of Computer Vision

The goal (Jahne, 2000) (Shapiro, 2020) of computer vision is to understand the content or features of the digital image. Understanding the content of digital images, involves the extraction of descriptors from the image. Further, this will help in developing methods that attempt to reproduce the capability of human vision. Typical goals of computer vision are:

- The detection, segmentation, localisation and recognition of objects present in the images.

- Tracking of the objects through an image sequence
- Mapping a scene to 3D modelling of the scene
- Estimation of 3D pose of humans and their limbs
- Content based image retrieval.

Computer vision performs differently if compared with image processing. Image processing focuses on enhancing the image by performing transformations like sharpening of the image, smoothing, noise removal, enhancing the colour contrast etc. Whereas computer vision focuses on extracting information from the input images to get the proper understanding of the input. It tries to mimic the cortex and predict the visual input like human brain does. Image processing uses various filtering methods, hidden Markov model etc. for the processing of the image, on the other hand, computer vision uses image processing as one of the methods, used with ML as well as DLmodule.

1.3 Computer Vision: Major Applications Areas

Computer vision aims to generate useful and intelligent descriptors information from the visual scenes by performing some operations on the signal received from the video cameras. It is been widely used in following areas (Jahne,2000):

- **Automatic Face Recognition:** Face detection is the foremost step in face recognition (Jain and Li, 2011) pipeline. It is the problem of locating and localizing the face present in the visual scene or photograph. Locating a face refers to finding out the co-ordinates of face present in the image, whereas localization often uses rectangular boxes to demarcate the region of face. Dynamic nature of faces makes it challenging problem for computer vision. Variation in light, occluded background, clothing, accessories like presence of spectacles, hair colour, makeup and aging are the common issues with detection system. In addition to face detection, face recognition is also gaining more attention in various areas where subject identity is required. It is the process of recognizing or verify the identity of the person from the stored database. One of the major application areas is security and surveillance. Also, it is helpful in analysing the expressions of the person.

- **Autonomous vehicle for visual guidance:** Autonomous vehicle (Umbaugh, 2010) requires more precise visual training to detect and recognize the presence of any object to avoid Collison. Localization, mapping and object tracking are the major primitives for any autonomous vehicle. The process of path planning and guidance depends on theses primitives.

- **Automated medical image analysis and diagnosis:** Computer vision has shown a tremendous improvement in the field of medical. Medical images like X-ray, CT scans, MRIs and ultrasound are used to develop the AI-based solution for detecting the abnormalities in the functioning of various organs (Gao, Yang, Lin and Park, 2018). Further, recent developments help the doctors for better understanding of the image by converting into 3D and allow easy interpretation.

- **Optical Character Reader (OCR):** It is a field of research in pattern recognition, artificial intelligence and computer vision (Jahne,2000). This tool is used to convert the scanned document or photo into text. It is widely used where paper records like passport documents, invoices, bank statements etc. are to be stored in the form of text for the analysis purpose.

- **Agricultural Industry for visual analysis and grading:** AI in agriculture sector is enhancing the performance by using powerful computer vision technology. AI enabled technique visually analyse the condition and fructify level to determine the time for harvesting with the reaping capability of such crops (Rosenfield,1997). Due to the high labour cost and unavailability of man power, high demand of agricultural products raises the need, farmers to adopt this latest automation technology. Further, that will help in minimizing the production cost and improve the crop yield with better efficiency.

- **Tracking of human and object:** The real-time detection of humans is evolving as a significant trend. The detection of human can be advantageous in areas like simple retail shop to surveillance applications (Jahne, 2000). The technology enabled cameras are able to detect the number of pedestrians walking on the road, number of visitors/intruders are going inside the office/house etc.

- **Visual biometric-based person identification:** Biometric systems are also known as realistic authentication system. It measures the body features and human characteristics. It is used to recognize and identify humans based on their biometric traits such as face, finger print, iris, gait etc. With the increase in security demands, the need of the technology is gaining more attention (Umbaugh,2010).
- **Anomaly Detection system:** Anomaly detection (Sultani, 2018, Nasaruddin et al. 2020) is the process of identifying the outliers in the present scenario. The outliers can be unusual point lying in the dataset, can be the instance where person is not following the trend. It can be widely used in the applications like fraud detection, health monitoring, event detection, intrusion detection etc.
- **Scene Understanding:** Scene understanding (Jahne,2000) is a most challenging problem of computer vision. The aim of the application is to understand and build a representation of the content present in the picture, like it gives the details of number and type of objects present in the picture and their relationships with each other. In case of presence of human being, it provides the details of what actions are being performed and what is the place and any other clue that can be advantageous for surveillance and monitoring applications.

1.4 Computer Vision: For Security and Safety applications

Traditional security system relies on human security guard for continuous monitoring and providing the security. Manual surveillance requires lot of efforts for continuous monitoring and observing. The accuracy of the result directly depends on the efficiency of the human being who is involved in guarding. These traditional systems suffer from limitations like Fatigue, irritation in eyes, biased nature of the person guarding.

Now-a-days video surveillance system are rapidly deployed in public spaces to strengthen public safety. Availability of powerful computing hardware at low cost motivates the use of technology for the benefit of society. In view of this, the limitation of the existing system and the demand of present world, poses the need to move towards more sophisticated automated surveillance system.

In recent years, the use and deployment of video surveillance has substantially increased worldwide. From figure 1.1 it is observed since 2015 uptil now and in the future also, the need and deployment of these systems will be in great demand. The market size has also increased with time and is expected to reach USD 33.60 by 2026. The statistics in figure 1.1 clearly suggest an uproar in the market of automated surveillance with the use of latest AI enabled technologies.

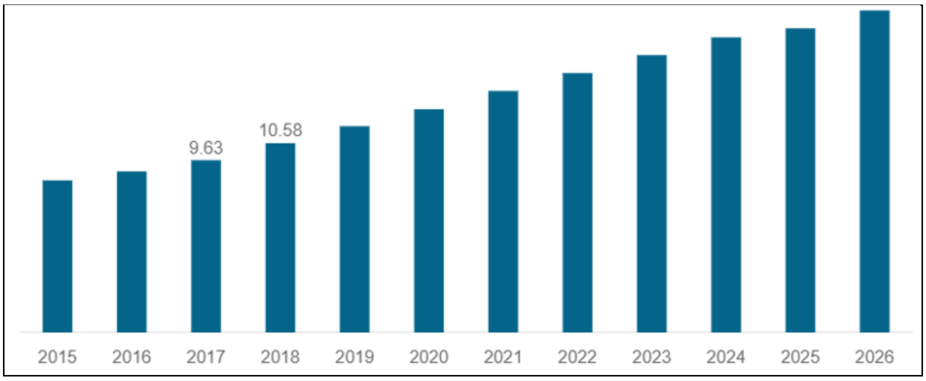


Figure 1.1: Asia Pacific Video Surveillance Market Size, 2018(USD Billions) (Fortune Business Insights)

The demand of video surveillance is increasing day by day. The market is divided into commercial, industrial, residential and government sector. As per the analysis by (fortunebusinessinsights.com), a major portion is held by commercial sector. The commercial sector has various industries like IT, telecommunications, retail and healthcare. Different sectors leverage the surveillance system as per the requirement of the field. Retailers use the surveillance system to analyse the activities of the customers throughout the store. It enables the system to detect unusual or abnormal event. Also, provides the solution for the store security. Moreover, the functionality can be extended to enhance the store management, loitering detection etc. Video surveillance provides a better and safer workplace, provides customer analytics and offers a superior safety performance. These factors enhance the need and use of video surveillance in retail sector.

The number of thefts and frauds transactions are increasing on a higher pace in various banks and financial institutions. The increase in ATM thefts and suspicious withdrawals across the world are driving the need of adequate security systems. Hence, deployment of video surveillance is necessary.

Surveillance cameras are gaining attention in healthcare industry also. Hospitals that have embraced the network-based surveillance system, intended to combine the visual data with audio. This integration enables the hospitals to extend the capacity of surveillance cameras to capture real-time data to generate real-time alerts in case of any emergency situation.

In universities, college and schools' security of the students is at utmost priority. Campus security cameras are deployed to identify the suspicious behaviour of the student. Also, it provides unauthorized access in campus. It monitors the activities of the students and any theft or crime activity can be prevented. Archived surveillance footages can be used to identify the individual involved in swindling act.

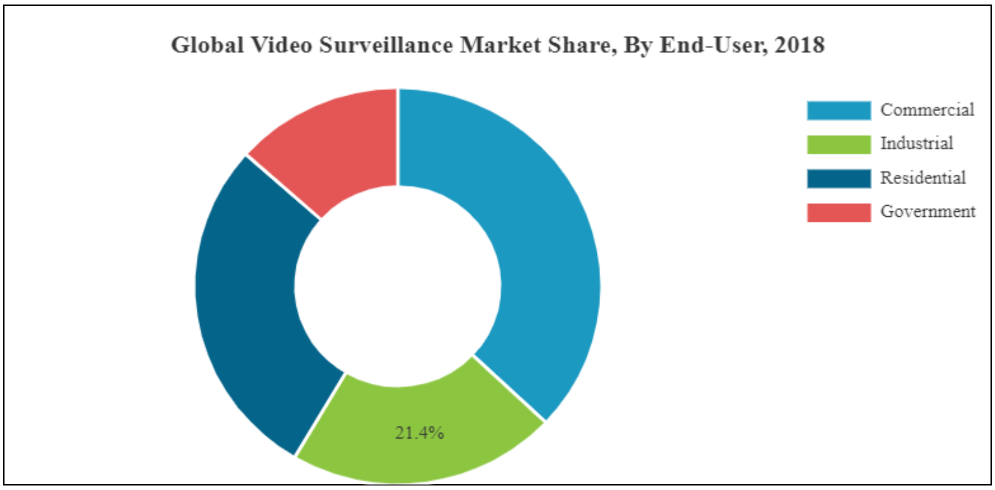


Figure 1.2: Video Surveillance Market Share

Other industrial sectors, figure 1.2, like manufacturing, construction and automobile etc. require the deployment of surveillance cameras. These sectors are also prone to security issues like vandalism, theft of heavy and expensive machinery raise the need of deployment of high graded surveillance cameras.

In residential sectors, surveillance cameras can be deployed to monitor the health of elderly people. The automated system will analyse the condition of the person and raise the alarm in case of any emergency.

1.4.1 Human Activity Recognition: Behavioural Perspective

To anticipate future actions in a real-time, is a key component of intelligence in various applications. Predicting human actions has a variety of applications in human-robot interaction, automated system that sense the environment and explore abnormal situations for security purpose, activity awareness for monitoring person’s health.

Human behaviour understanding is a very challenging task. It aims at determining and recognizing person’s activities based on sensor and/or video observation-based methods. The main goal of HAR (Vrigas,2015) systems is to observe and analyse human activities and to interpret the ongoing events. Using approaches like vision-based and sensor-based method (Dang,2020), HAR system retrieve and process the contextual information to understand the person’s behaviour. Contextual information can be related to objects present in the environment, spatial and temporal movement.

It is gaining more attention due to applications in smart video surveillance, ambient intelligence and content-based information retrieval (Vrigkas, Nikou & Kakadiaris, 2015). Human Behaviour is a complex interplay of three components: Actions, Cognitions and Emotions. For further consideration, human actions are considered for behaviour analysis.

Security is an emerging paradigm which has a great importance in almost every field. The purpose of human security is to prevent the occurrence of some risk or threat. Such areas require early detection of mischievous activities, so that a big threat can be avoided. In order to obtain security, CCTV cameras are installed for the monitoring of human activity. Capturing intent involves recognizing facial expressions, gestures, and activity from the CCTV footage to predict the intention of the human. A gesture is a biomechanical movement whereas an activity involves a full-body movement.

Recognizing which actions are being performed within a context is a key to determining the probable intentions of the individual involved.

Activity recognition involves tracking individuals over time as they perform a series of actions. By monitoring the trajectory of a person, the involvement of a person could be identified in some swindling act.

In general, HAR process involves following steps to accomplish the task (Sargano, Angelov &Habib,2017). These are:

1) Data Collection
2) Pre-processing of the data involves noise reduction, image quality enhancement etc.
3) Feature extraction corresponds to extracting of spatial and temporal features from the given data.
4) Classification

1.4.2 Human Activity Recognition: Surveillance as an Application

With the increase in crime rate, there is a massive increase in the installation of surveillance cameras. In order to ensure public safety, various CCTVs have been installed in the areas like airport, railway station, colleges, schools, day-cares, malls, parks and other public places. Monitoring these surveillance cameras on 24x7 basis, requires a lot of manpower. The accuracy of the system depends upon the activeness of the supervising entity. Sometimes, biased nature of the supervising person, fatigue issue results in the inaccurate decision. Therefore, there is a dire need of automated surveillance system, that can detect the anomalous event correctly and in time. Detection of anomalous event (Sultani & Chen,2018) from surveillance videos is a very challenging problem of computer vision domain. The complexity and size of the data, illumination variation, occlusion, noisy environment etc. makes the problem more challenging.

The problem of detecting anomalous event is actually recognizing the human activity closely. HAR is gaining more attention in other domains like healthcare, education, Ambient Assisted Living (AAL), Human-Computer Interaction (HCI) and many more. HAR is related with understanding of the actions posed by human being. Activity is the collection and sequence of actions performed by human being. The human activities are categorized into following levels:

- Gesture - A gesture is a primitive movement of any body part to convey some information. It can be a small hand movement or just the facial expression. The duration for such activities is very small.

- Atomic Action - it is a simple activity (involves several gestures) poses by a human being. Examples are like: jogging, running, swimming.

- Interaction - it is an activity between human and another agent. The other agent involved in interaction can be a human being or any object. If the interaction is in between human and human then it is termed as human-human interaction. Examples of such interactions are: shaking hands, hugging each other. And if one of the agents in interaction is any object then it is human to object interaction like a person is using his laptop.

- Group Actions- it is a complex activity which involves more than two human and objects. Example of group activities are: playing a game in a team like football, playing cards in a group, group fighting, parade etc.

Figure 1.3 shows, as we move around the axis, the complexity level of the human activities also gets increased. Hence, difficulty to automate the process also reaches a certain extent.

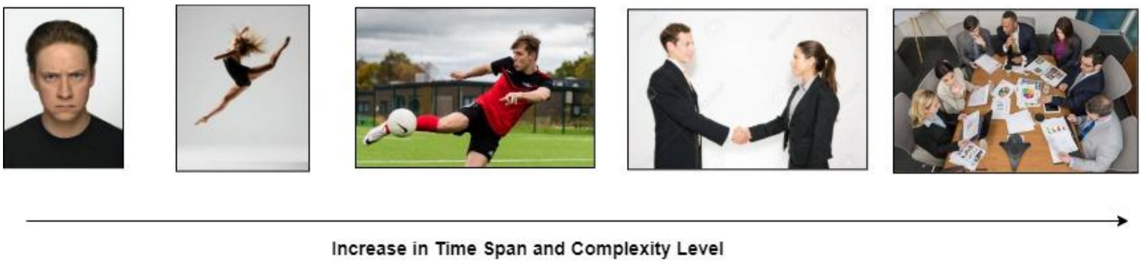


Figure 1.3: Levels of Human Activities

The working of HAR is mainly divided into 4 phases: Data collection, Pre-Processing, feature extraction and pattern/ activity recognition. Data Collection is a foremost task for the accomplishment of defined goal. Collected data is pre-processed to feed into the model. Further, extraction of features plays an important role in the success of HAR. Various algorithms have been developed based on local and global feature descriptor based on single to fusion of multi-feature descriptor. Further, the pattern analysis is important to understand the activity performed.

1.4.3 Challenges

HAR through visual sensor networks is one of the popular approaches of computer vision. Sensors have the capability to gather and process the information from the environment. The major difference between the visual sensor and other sensors, is the way it perceives the information. Most of the sensor data is represented in 1-dimensional whereas, visual sensor provides the data in 2-dimensional. This 2-dimensional data which is stored as image frames, are helpful in analysing the spatial features. It is very challenging to handle temporal structure of videos without compromising the performance of the model, with somewhat reasonable parameters and complexity.

1.5 Machine Learning: A Deep Learning Perspective

ML which is a sub branch of artificial intelligence intern has a sub branch called DL. It deals with creating ML models which in some form mimic the human brain. This form is called a Neural Network. A neural network, just like the human brain is made up of little building blocks called the Neuron. While a computer neuron and an actual neuron share the same name, their form is very different. A computer neuron is a node which can take continuous values from zero for being completely turned off (not active at all) to one for being completely turned on (fully active).

A neural network (Abiodun, Jantan, Omolara, Dada, Mohamed and Arshad, 2018) is made of layers which are further made up of multiple neurons which can be thought of as lined up in a straight line. These layers can be divided into three categories namely The Input layer, hidden layer and output layer. They are named so as the input layer of the network takes the input of the problem, the output layer gives the output to the problem and the hidden layers while not truly hidden performs calculations which can be hidden to the rest of the program without making any difference. Each neuron present in a layer has a direct connection to each neuron in the immediate previous and immediate next layer starting from the input layer to the output layer. Each of these connections has a certain weight and a certain bias that decides how much the activation of a current neuron will contribute to the neuron in the next layer that the previous neuron is connected to. This series of activations starting from the input layer (input layer usually has activation of neurons as either zero or one) of neurons passed down through layers via connections ultimately finds its way to the output layer, where a certain neuron or a set of neurons being active delivers a meaning and a different set of neurons has a different meaning depending upon the problem structure. Each of these parameters when changed has an impact on how well or bad the network performs.

1.5.1 Need and Advantage

ML algorithms which are not DL algorithms had three major drawbacks which lead to the opening of a pit which was later filled by DL algorithms. These drawbacks were: -

1) Slow performance at the time of prediction.
2) There is a need for feature extraction and scaling.
3) The performance of non-deep learning algorithms grows towards a plateau as the data size increases.

The advantages that DL (Goodfellow,2016) holds over other algorithms are both due to intrinsic properties and due to the type of hardware we have access to these days.

Most problems dealt by Artificial Intelligence and its sub branches fall in the NP complexity category and these problems by their innate nature are difficult to solve. All the algorithms that do tackle such problems are more or less complex and take time to run. Even a few decades ago training any ML algorithm was a really time-consuming task, however with the recent performance boost provided by new and specialized processors and other hardware training neural networks has become significantly easy. Now that training neural networks is easy, it has been observed that running a DL algorithm is faster to yield results than one that is not a DL algorithm.

DL algorithms do not require manual feature extraction or scaling. This is by the virtue of its structure and the way a neural network learns. The weights and biases of the network set themselves in a manner so that they adjust any feature that hasn't been scaled to the other features. Moreover, any feature extracting that has been left out is also taken care of.

Another advantage that a DL algorithm has over a non-deep learning algorithm is that with the increase in training data the weights and biases keep on changing to fine tune the network to increase its performance. Whereas in a non-deep learning algorithm the model after a certain limit reaches a plateau in terms of performance even if the training data is increased thus converging to certain constant upper limits of performance.

1.6 Deep Learning Models: Activity Recognition

Capability of DL can be extended for many tasks and one such particularly difficult task is activity recognition, specifically HAR. A seemingly easy task for humans, to identify whether a person is sitting, standing, running or something else becomes a really difficult task for machines to do even with access to a lot of data for training. It requires a lot of feature extraction to find particular data patterns to look for to recognize and distinguish between different activities. For this reason alone, it becomes a difficult task, as a human is required to first interpret the data and then pull features out of it for the model to learn from and this leaves room for errors and imperfect models. This is where DL comes into play, specifically because of their ability of learning without the need of feature extraction, neural networks perform quite well at recognizing patterns from raw datasets to recognize and differentiate between activities.

Traditional ML based models relies on handcrafted (Dang,2020) and human knowledge to extract features from the given data. These methods are incapable to identify the complex features from the data, collected from sensor-based and vision-based approaches. Recently, DL-based methods have shown a great performance in the field of HAR. These methods have capability to extract the complex features by its own. Convolutional Neural Network (CNN) (Albawi, Mohammed & AI, 2017) and Recurrent Neural Network (RNN) (Murad & Pyun,2017) have delivered promising results in recent times.

DL methods are based on neural networks, having multiple processing layers. These layers provide the level of abstraction and useful for extracting the required representation. In addition, more complex, high level and more meaningful features can be learned through DL-based approach. Furthermore, deep neural networks are more suitable for unsupervised data. This approach of using deep structure is ideal for HAR. Idea of DL-based methods are discussed in sub-section1.6.1 and 1.6.2

1.6.1 CNN

Convolutional Neural Network (CNN) is a multilayer neural network, inspired by the visual perception of the human beings. It became popular after the tremendous performance of Alexnet (Krizhevsky, Sutskever, and Hinton,2012) in 2012 in Image recognition challenge. It is mainly popular and widely used in the area of Image Recognition. It has capability of modelling 2D and 3D images. The structure of CNN is composed of: 1) convolutional layer 2) pooling layer and, 3) fully connected layer (FC).

The Convolutional Layer is the building block of a CNN. The aim of this layer is to learn features from the given input. Varying size filters/kernels are used to compute various feature maps. The generation of feature map is dependent on the convolving operation performed on input with kernels and applying elementwise non-linear activation function on the convolved result. The task of sub-sampling layer is to select a small region of the convolutional output as input and produce a single output on down-sampling. Various techniques of sub-sampling like max pooling, average pooling, min-max pooling etc are available. Purpose of pooling is to reduce number of parameters to be computed. The final part of CNN consists of Fully Connected (FC) layer. It takes input from the pooling layer and generates final output.

1.6.2 RNN

Unlike CNN, Recurrent Neural Network (RNN) is not a method built to process data before feeding it to the network, it is a neural network with a different structure. Unlike a conventional neural network which is directed only in one direction a recurrent neural network has some of the neurons in layers deeper in the network affecting neurons in the previous layers. This loop of layers affecting previous layers, while complicating the structure, creates internal memory which helps reduce the parameters (the weights and biases) required to tune the network's performance. This internal memory structure (Singh, Merdivan, Psychoula, Kropf, Hanke, Geist and Holzinger,2017) allows the network to process input strings of variable lengths that along with their complex structure allows them to be quite efficient at tasks which other ML models struggle at, such as machine translation, speech recognition, understanding language with context and grammar and many others.

RNN broadly has two sub categories namely finite impulse RNN and infinite impulse RNN. Both being more or less the same have one major difference, i.e., the connections between the neurons in finite impulse RNN do not form closed loops (infinite cycles) whereas the ones in infinite impulse RNN do. This means that if stretched a finite impulse RNN can be converted to strictly forward feeding RNN structure although the RNN formed would have connections skipping layers and moving to layers not directly in front of them.

1.7 Building Blocks of Deep Learning Models

1.7.1 Activation Function

Activation functions are the most important and decision-making parameter of neural network (Nwankpa, Ijomah, Gachagan and Marshall,2018). They are used to add non-linearity to the model and to learn complex relationships of the variables. It is applied on the hidden layer as well as on the output layer. The selection of a right activation function helps further in learning process. The selected activation function is applied on the output of a convolution, and returns a transformed output. There are two types of activation functions are available:

1) Linear Activation Function: This is the simplest form of activation function. A network comprised of single layer with linear activation function are easy to train. But they are unable to learn the complex features of the data. They are mostly used in the output layer.

2) Non-Linear Activation Function: Non-linear functions are used to learn the complex mapping of the data. Different activation functions are used in the literature. Linear, sigmoid, tanh, ReLU, leaky ReLU, maxout, etc. are most commonly used activation functions.

Non-Linear Activation Functions

Deep Networks uses activation functions to perform the complex computations in the hidden layers. The hidden layer's output is transferred to the final output layer. The purpose of using activation function is to add non linearity to the network, so that complex features learning can be adopted. Following are the various activation functions:

1) Sigmoid Function

Sigmoid function (Sibi, Jones and Siddarth, 2013) is basically the probabilistic approach of decision making. It accepts real numbers as input and produces the output in between 0 and 1. The curve of the sigmoid looks like S-shape, given in the figure 1.4. The equation for the sigmoid function is given as:

$$S(x) = \frac{1}{1+e^{-x}} \tag{1.6}$$

In literature, sigmoid was very widely used activation function. But now, a day its application is diminished because of the reason - The sigmoid function causes a problem of vanishing gradients. Whenever the output function is in the tail of 0 and 1, the value of gradient function becomes almost to 0. During the backpropagation phase, this low value of gradient will affect the parameters destined for the objective function.

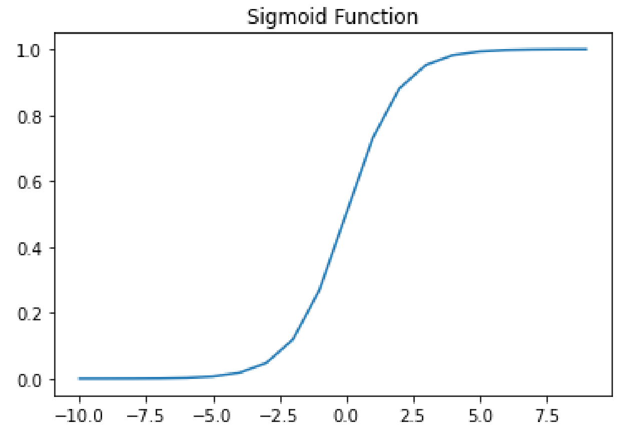


Figure 1.4: Sigmoid Curve

2) Tanh Function

Tanh Function (Sharma, Sharma and Athaiya,2017) is another type of activation function. It has similar capabilities like sigmoid function except it binds the output in the range from -1 to +1.

Tanh function equation is given as:

$$T(x) = e^{x} - e^{-x}/e^{x} + e^{-x} \tag{1.7}$$

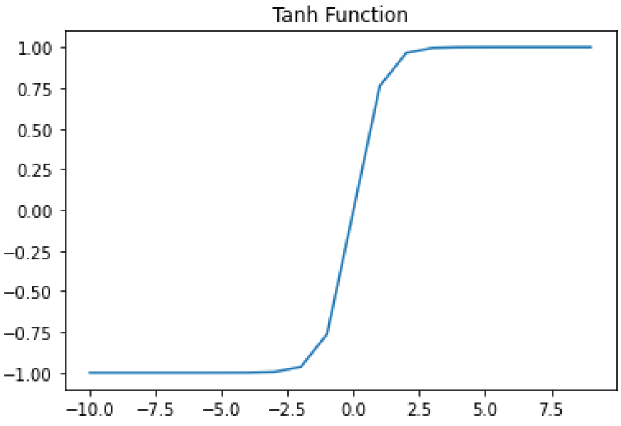


Figure 1.5: Tanh Curve

The advantage of using tanh is that the produced output is zero-centered and gradients are not forced to move in a particular direction. Figure 1.5 shows the curve of Tanh function.

3) ReLU Function

The Rectifier Linear Unit (ReLU) (Nwankpa, Ijomah, Gachagan, and Marshall, 2018) is the widely used activation function for DL models. The mathematical representation of ReLU is:

$$R(x) = \max(0, x) \tag{1.8}$$

This function returns value ‘0’ if it receives the negative input and for any positive value it returns that value back as the result as shown in figure 1.6.

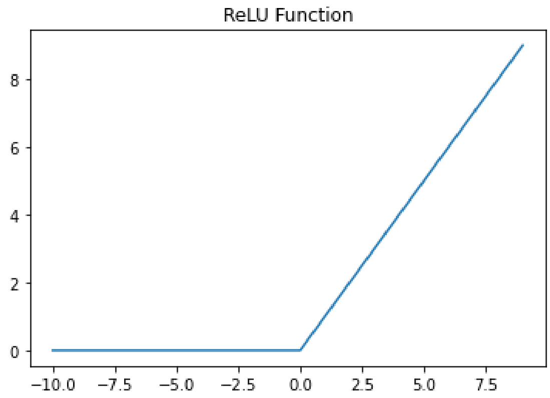


Figure 1.6: ReLU Curve

The advantage of using ReLU over other activation function is that it doesn’t deactivate or activate all the neurons at the same time. From the above figure: it is clearly understood that if the output of the transformation is less than 0 then only the neurons will be deactivated. Hence, required less computational load. But the issue with the ReLU function is that it considers all the non-positive values to 0. That results in poor mapping of negative values. It is known to be *Dying ReLU situation*. To solve this problem; variant of ReLU is available:

Leaky ReLU

It is the improved version of ReLU activation function. Unlike in ReLU, it doesn’t ignore the negative inputs completely. The mathematical representation is given as:

$$f(x) = \begin{cases} \varepsilon . x & for\ x < 0 \\ x & for\ x \geq 0 \end{cases} \tag{1.9}$$

Where, Ɛ is a constant, usually set to a small value like 0.001

Softmax Function

It is the also a probabilistic approach like sigmoid (Sharma, Sharma and Athaiya,2017). The difference between sigmoid and softmax is that sigmoid is mainly used for two class classification problem whereas the later one is applicable for multiclass classification problem. The mathematical representation of Softmax is as:

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_j \exp(x_j)} \tag{1.10}$$

This function distributes the probability throughout among every node present in the output layer. This function transforms the input into output in such a way that the total sum of the probability lies in between 0 and 1.

1.7.2 Loss Function

Loss Functions are integral part of any DL-based model. For each problem, an objective is defined and against which the performance of the model is measured. Thereafter, weights and bias parameters are adjusted with the aim to minimize the error/loss function.

The last layer of CNN model is the output layer, which is responsible for the final classification. This layer uses a loss function also known as cost function to estimate the predicted result. This layer predicts the output and prediction error/ loss is computed over training samples by CNN model. This can be computed through various loss functions. It provides the information about how much predicted output is deviated from the actual output. This difference is computed and model tries to optimize the performance during learning process. There are various types of loss functions used in different domains of problems. Loss functions (Gu, Wang, Kuen et.al ,2018) are categories in two broad categories: 1) Regression Loss 2) Classification Loss.

Most common functions are briefly described here:

Cross Entropy Loss Function: Cross entropy loss is also known as log loss. It measures the performance of the model whose output is a probability which value belongs to 0,1. The value of loss increases when the predicted result diverges from the actual result. This function is widely used for the calculation of loss. It is preferred over mean squared error because it uses logarithm function, log value offers penalty for small differences between predicted and corrected probability. When the difference between two is observed large, penalty will be higher. For two class problem, binary cross entropy is used whereas for multiclass problem, categorical cross entropy is recommended. Cross Entropy Loss can be computed as by the equation given in equation 1.11,

$$loss = \ -\sum_{i=1}^{C} t_i \, log(p_i) \qquad (1.11)$$

where t_i is the true label and p_i is the SoftMax probability for i^{th} class, in a model of total number of C classes.

Euclidean Loss Function: Euclidean loss is also known as quadratic and mean squared error loss. The loss function layer over an input Xi computes the deviation of Xi from target Ti using Equation 1.11. The output of this layer is the amount of error computed and is not passed further to any other layer in the network. So, the Yi is non-existent. Further, the Euclidian loss function is not characterised by any weight w.

$$Error = E = \frac{1}{2}\sum_{i=1}^{m}(x_i - t_i)^2 \qquad (1.12)$$

Hinge Loss Function: Hinge loss is used for computing the loss where input vector ‘x’ labels as ‘y’ Target values lies in the range between [-1, 1], which makes it best for binary classification. It is mainly used in case of support vector machine, where maximum-margin classification is required.

1.7.3 Role of Optimizers and types

Optimizers are algorithms that are used to minimize the error or loss function. They are the mathematical functions that tries to reduce the loss function by making adjustments in the learnable parameters i.e., weights and bias.

Gradient Descent*:* This is the simplest approach used for training neural networks. This optimization algorithm based on a convex function.

Gradient Descent (Schmidhuber, 2015) is a technique to minimize the objective function $J(\theta)$ by updating the parameters in the opposite direction of the objective function $\nabla_\theta J(\theta)$ with respect to all the parameters.

It adjusts the parameters iteratively with the aim to reach local minimum. Gradients are partial derivatives of the loss function and effects the resultant loss. It moves in the direction opposite to the steepest ascent. The step size, in order to achieve local minima, is decided by learning rate (η).

Too high value of learning rate gives abrupt behaviour, whereas too low value takes multiple iteration to reach minimum point. So, it is important to carefully choose the value of learning rate.

Gradient Descent uses the complete data of training set for calculating the gradients, which makes it slower and computationally expensive. Following approaches are present for the computation of gradients:

- ***Batch Gradient Descent:*** It is the simple and straightforward approach to calculate the gradients. Considering all the parameters of the entire dataset, this approach computes the cost function. This approach is very slow, because to perform just one update, gradient calculation for complete dataset is required.

- ***Stochastic Gradient Descent (SGD):*** It is a variant of gradient descent, which focus on regular operation of parameters with each iteration. Training loss is computed after executing each training sample and parameters are updated. SGD method focus on frequent updates and hence reaches the minima in less time. It is computationally less expensive as well as demands low memory requirements.

- ***Mini-Batch Gradient Descent:*** This is another variant of gradient based method. In this approach, a mini-batch of samples is selected from the entire dataset. Selected subset is used to compute mean gradient and to update model parameters with the aim to reach local minima. This approach is computationally faster and requires less memory space compared to other techniques discussed above.

Optimization Algorithm

There are various optimization algorithms (Goodfellow, Bengio and Courville, 2016) that are used by DL methods. Momentum, Adaptive Gradient Algorithm (Adagrad), Adam, RMSProp are discussed further.

Momentum: It is an algorithm which performs on the basis of exponentially weighted average of all the gradients. This average makes the algorithm to converge towards local minima at a faster speed. From the figure 1.7 (a), it is observed that SGD has difficulty navigating ravines, to reach local minima (shown by red dot). It is oscillating across the slope of ravine rather moving horizontally towards local minima. Momentum is a technique which helps SGD to accelerate in the relevant direction as given in the figure 1.7(b). This happens because of the fraction γ, it is added to the update vector of previous time step shown in equation 1.12. The value of momentum term γ is usually chosen between 0 and 1. In general, it is empirically suggested to use the value of γ to be 0.9.

$$m_t = \gamma . m_{t-1} + \eta \nabla_\theta J(\theta) \tag{1.12}$$

$$\theta = \theta - v_t$$

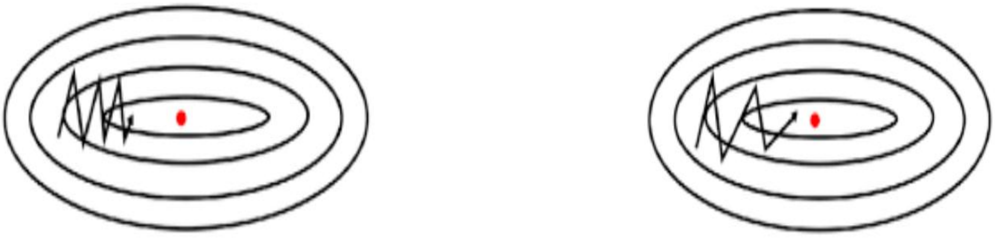


Figure 1.7 SGD without Momentum (left) SGD with momentum(right)

Adaptive Gradient Algorithm (Adagrad): It is stochastic optimization method that changes the learning rate (η) for each parameter (θ) at time step (t). The change in the learning rate is always performed on the past gradient for $\theta_{i.}$ It works on the derivate of an error function. Changing the learning rate affects the pace of training. For sparse feature input, a higher learning rate can be adopted which helps in boosting the dying gradient result. Whereas, on dense data, slower learning or low value of learning rate can be chosen. Equation 1.13 gives the per-parameter update which is vectorize further. $g_{t,i}$ is set as the gradient of the objective function w.r.t each parameter θ_i at each time step t_i.

$$g_{t,i=} \nabla_\theta J(\theta_{t,i}) \tag{1.13}$$

Whereas, g_t denotes gradient at time step t.

$g_{t,i}$ is the partial derivative of the objective function w.r.t to the parameter θ_i at time step t

The SGD update for every parameter θ_i at each time step t is calculated as in equation:

$$\theta_{i+1,i} = \theta_{t,i} - \eta . g_{t,i} \tag{1.14}$$

In its update rule, Adagrad modifies the learning rate at each time step for every parameter based on past gradients that have been computed earlier for each parameter as:

$$\theta_{i+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \mathcal{E}}} (matrix\ vector\ product) g_{t,i} \tag{1.15}$$

In equation above, $G_t$is a diagonal matrix where each diagonal element i,i is the sum of squares of the gradients w.r.t θ_i. Ɛ is the smoothing term that avoids division by 0.

As G_t contains the sum of squares of the previous gradients with respect to all the parameters along the diagonal. Element wise matrix vector product is computed between G_t and g_t. The final update rule is computed by the equation:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \mathcal{E}}} (matrix\ vector\ product) g_t \tag{1.16}$$

The benefit of using AdaGrad is that it eliminates the need to manually tune the learning rate. Also, it tries to offer this adaptiveness by decaying the learning rate in proportion to the updated history of the gradients. Whenever there are larger updates, the history element is accumulated But, its ability to handle the squared gradients in denominator is weak. Since, every added term is positive and the resultant sum keeps on increasing during the training period. That accumulated increasing sum cause the learning rate to shrink, which becomes very small at some instance.

Root Mean Squared Propagation (RMSProp): It is a gradient based optimization technique that is used for training neural network. Gradients of very complex functions has tendency to vanish or explode. RMSProp was proposed by Geoffrey Hinton, with the aim to handle gradient problem. This is an improvement over AdaGrad optimizer. It handles the issue by taking the exponential average of the gradients instead of taking the cumulative sum of squared gradients. This normalization balances the step size, decreases the step for large gradients to avoid exploding and increases the step for small gradients to avoid vanishing. In short, this algorithm prevents the learning rate decay from diminishing too slowly or too fast.

Adam Adaptive Estimation (Adam): It is also used to compute adaptive learning rate for each parameter. It extends the capability of ***RMSprop*** and ***momentum-based gradient descent***. Like RMSprop, this algorithm uses the squared gradients to scale the learning rate. Also, it considers the average of the gradients instead of gradient itself like SGD with momentum.

Adam optimizer uses the strength of mentioned two algorithms and tries to find local minima with minimum number of oscillations.

$$m_t \ = \beta_1 m_{t-1} + (1 - \beta_1).\, g_t \tag{1.17}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2).\, g_t^2 \tag{1.18}$$

In above equations 1.17 and 1.18, m_t and v_t represents moving averages, g_t represents gradients on current mini-batch and β_1 and β_2 are the decaying rates of average of gradients and square of gradients respectively.

1.7.4 Hyperparameters

Hyperparameters (Goodfellow, Bengio and Courville, 2016) has a huge impact on the model's performance. There are variety of parameters like learning rate, batch size, number of epochs etc. are required for the training purpose. Not all, but some of the parameters are updated in order to attain better accuracy of the model.

Learning Rate: Learning rate is a parameter which affects the training time and accuracy of the model. It is used in such a way that will optimize the model's capacity. Learning rate selection is very challenging task. It is a scalar, value that tells the machine how fast or how slow it has to converge towards the result. Model training speed is important and varies with different applications. Sometimes, a very low value results in long training process whereas larger value speed up the process and henceforth missed out some important corelations. Hence, results in wrong classification.

Batch Size: It is related with the count of training samples to be used for the training purpose. Ideally, batch size must be > 1 and should be <= to the size of training dataset size. It has a great impact on accuracy of the model. Small batch size has ability to generalize better as compared to large batch size. And sometimes, large batch size yields in good accuracy but poses a constraint to computational cost.

Number of epochs: Epochs are the number of iterations that a model has to undergo on whole dataset. It can be defined with any integer value. The value can be set from one to infinity. Early stopping is a technique that is used to stop the training process. For example, on using combination of validation and training error, model training is stopped at the stage when the error noted is above the threshold value. In such cases, early stopping is the technique which could help in saving the computational time.

Major Concern

There exist various factors which need to be considered for video classification tasks. These factors help in designing and opting a network architecture. Video data has spatial and temporal domains to be considered. There are other important aspects to be taken care are as follows:

- ***Computational Complexity:*** It is defined by the number of operations required by the network architecture to perform certain task. Number of layers and types affects the performance of the model.

- ***Computational Speed:*** If the model requires large number of complex operations to obtain the output, will result in the degradation of the computational speed. It is strongly related with the computational complexity.

- ***Accuracy:*** Accuracy is the crucial factor to determine whether the model is able to give a feasible solution for the given problem.

1.8 Objective

One of the major issues in 21st century is increase in crime rate that caused loss of life and property. Surveillance system is one of the most favoured solutions for early detection of anomalous events/abnormal events. For this, a massive installation is done everywhere. The accuracy of this traditional system relies on the activeness of the supervising entity. Sometimes biased nature, fatigue issue may report in intolerable results. Therefore, there is a dire need of designing an automated surveillance system with the capability of predicting the ongoing event or suspicious behaviour of the person.

In the current scenario, to design a HAR system, following objectives are proposed:

1) To study the concept and underlying principles of HAR system and its importance in the area of automated surveillance system.

2) To understand the need and importance of DL in the area of computer vision.

3) To introduce a new deep learning-based approach, which will be able to learn and classify scenes with anomalies from surveillance videos.

4) To compare the performance of the proposed model with the existing models present in this domain.

5) To evaluate the aforementioned objectives.

1.9 Block Diagram of the Proposed Work

The work across the chapters of a thesis explains the basis of the proposed system. In order to understand the concept of the proposed system a basic block diagram of the proposed work is presented here. Figure 1.8 shows the conceptual flow graph of the work.

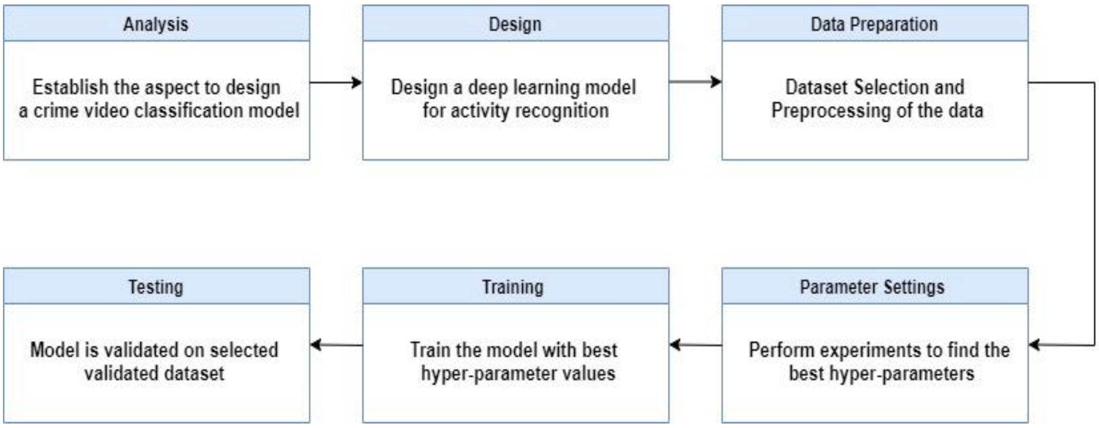

Analysis
Establish the aspect to design a crime video classification model
Design
Design a deep learning model for activity recognition
Data Preparation
Dataset Selection and Preprocessing of the data
Parameter Settings
Perform experiments to find the best hyper-parameters
Training
Train the model with best hyper-parameter values
Testing
Model is validated on selected validated dataset

Figure 1.8: Block Diagram of the Proposed Work

1.10 Organization of Thesis

This work is structured into 6 main chapters named as follows: Chapter 1-Introduction, Chapter 2-Literature Survey, Chapter-3 Deep Learning evolution for Image classification, Chapter-4 Inception and LSTM based Methodology for HAR, Chapter-5 Human Activity Recognition Model based on Motion Flow and Chapter -6 which summarizes the work done and future work.

1.11 Summary

The idea of conjoining computer vision with DL can become a good platform for understanding the HAR more effectively. System thus developed can be used for the welfare of the society and mankind. So, in this chapter, the focus is on understanding the basic fundamentals, challenges and the motivation behind the proposed model for HAR.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

2.1.1 Human Activity Recognition using Deep Learning

In recent decades, the computer vision field is focusing highly on one of the most essential problems: human activity analysis. It could be helpful for almost everywhere in the intelligent surveillance systems to bettering human-computer interactions. Recent techniques have been quite adept in identifying the individual activities. Moreover, HAR have gained prominence in public health activity monitoring (Ma, Chen, Kira and AlRegib, 2019) (Suresh and Visumathi, 2020). Generally, activity recognition is being used in a variety of technologies to keep track the daily activity routines of the humans. In recent years, several approaches have indeed been recognized and presented. Due to the extreme complexities of human facial vision and the substantial shifts caused by various influences; designing and implementation of the sophisticated computational system for video based HAR is more difficult. Employing supervised learning frequently needs considerable training, and hence results in longer execution times (Weng, Li and Jin, 2021) (Aghaei, Nazari and Moghaddam, 2021).

CNN and RNN models for Activity Recognition

ML based researches has achieved significant advancements in this field. After that, the DL approaches have brought out significant improvements over the ML models. In recent years, the CNNs and the RNNs are two sorts of DL models that have been frequently utilized to solve the HAR problem (Noori, Wallace, Uddin and Torresen, 2019) (Venkata Krishna Rao, Gopikrishna and SukritaLatha Parameswaran, 2021). Moreover, CNNs are increasingly being used as a feature learning approach for HAR. CNNs could automatically extract the features, unlike traditional ML approaches, which need domain-specific knowledge (Chowdhury., Tjondronegoro, Chandran and Trost, 2017) (Wang, Candinegara, Hou. Tan and Miao, 2017).

To categorize the activity data captured by smartphone sensors, Ronao *et al. (*Ronao and Cho, 2016) utilise a “1-Dimensional CNN”. CNNs could automatically extract the features, unlike traditional ML approaches, which need domain-specific knowledge. Researchers have analysed the accuracy of their developed framework with several shallow ML models, such as SVMs and DTs. Finally, CNN model appeared to be more reliable, according to the statistics. Zebin*et al.* (Zebin, Scully and Ozanyan, 2016) have categorized 6 regular chores collected from twelve participants using a 2-Dimensional CNN. In terms of consistency and processing overhead, the researchers compared their technique against standard ML algorithms. In terms of both metrics, the outcomes manifested more improvements. Ha *et al.* (Ha and Choi, 2016) have introduced another CNN-based technique for activity categorization that uses 2-Dimensional CNNs. Researchers have examined the performance of two CNN models that primarily differs in the way of weights redistributed in each convolution layer. Another CNN adaptation to HAR has been presented in (Chowdhury,Tjondronegoro, Chandran and Trost, 2017). The author's objective was to look at the various sensors configurations and recommend the appropriate sensor arrangement for lower-limb activities. The researchers of (Arifoglu and Bouchachia, 2017) have used many RNN versions (e.g., “GRUs and LSTMs”) to characterize everyday tasks and to detect the aberrant behavior in dementia sufferers. Researchers have compared the performance of these approaches to shallow ML models. The researchers therefore concluded that RNNs had surpassed other ML techniques on almost all of the examined metrics (e.g., “accuracy, precision, and recall”). The key objectives of HAR systems are the effective tracking and evaluation of human actions, and also to successfully interpret the ongoing occurrences. Moreover, the researchers have suggested that, the LSTM technique is more precise than CNN and RNN for activity recognition. HAR technologies collect and interpret contextual (environmental, geographical, temporal, and other) information using visual and non-visual sensory input to comprehend people's behavior. On the other hand, the Data collection is an essential element in HAR system, as the input data influences the overall performance.

Recognition of human activity seems to be important in human-to-human communication and personal relationships. It's indeed challenging to retrieve since it contains information about a person's identity, personality, and psychological condition. One of primary themes of research inside the scientific fields of computer vision and ML is indeed the human ability to identify another person's activity. Many applications, including as "video surveillance systems, human-computer interaction, and robots" for human behaviour characterization, currently necessitate a multiple activity detection system. Two key questions arise from different categorization techniques: “What action?” (i.e., the difficulty of recognition) and “Where in the video?” (i.e., the difficulty of localization). Whenever endeavouring to recognise human actions, one must also understand a person's kinetic states in order for the computer to recognise the activity effectively. Human activities including "walking" and "running" occur organically in everyday life and are very straightforward to identify. Particularly complicated actions, such as "peeling an apple," are, on the other hand, harder to comprehend. Sophisticated actions can indeed be broken down into smaller ones that are more easily recognised. Object recognition in a scenario may usually aid in properly understanding human actions by providing relevant information about the current occurrence.

2.1.2 Types of HAR

The input data fed to HAR systems are typically classified as: sensor-based HAR and vision-based HAR (Mustafa, Dhavale and Kuber, 2020) (Imran and Latif, 2020). Following that, each group is subdivided into subgroups that conduct various operations, such as data gathering, pre-processing methods, feature engineering, and training. It seems to be a tremendous progress in vision-based activity recognition (Peng, Huang, Chen, Zhang and Fang, 2020).

Vision Based

The vision-based activity recognition has been highly deployed in “Human-computer interaction, user interfaces design, robot learning, and surveillance as well”. ICCV and CVPR seem to be two scientific conferences in which vision-based activity recognition research is widely presented. RGB data and RGB-D data are indeed the different sources of data used in vision-based HAR research (Sarma, Chakraborty and Banerjee, 2019)

Sensor Based

Sensor-based activity recognition analyses a variety of human behaviours via integrating the burgeoning technology of sensor networks with innovative data mining and ML approaches. Mobile devices (such as smart phones) have quite enough sensor data and computational resources to distinguish the physical activity. Sensor-based HAR is being used in a variety of real-world settings, notably in home-automation and nursing (Ahmad, Kazmi and Ali, 2019) (Sarma, Chakraborty and Banerjee, 2019). Moreover, the rapid increase of WSN has resulted in a massive volume of data (including wearable sensors, object sensors, and environmental sensor). Researchers conducted on "sensor-based activity recognition" have suggested that only by allowing the ubiquitous computers and sensors to monitor the behavior of individuals (with their permission), the HAR system can be well-suited to different applications.

This chapter discusses about the recent works undergone in activity recognition and face detection. Among all the DL models, the LSTM and Inception are indeed the trending one, and hence we've focused heavily on these models. In addition, we've discussed the research gaps and challenges that have been faced by the existing models, while working with sensor –based data as well as vision-based data. The research gaps tend to be a milestone for the future researchers.

2.2 Hybrid Inception and LSTM for Behavioural Analysis

This section discusses about the recent works undergone in activity recognition as well as face detection in surveillance applications. In field of activity recognition, few studies have been conducted by combining the fuzzy techniques with ML solutions. These solutions weren't satisfactory, as their robustness and reliability were bit lower. Moreover, the fuzzy models require huge training data, which has automatically boosted up the system complexity. The CNN and RNN models were good in HAR, but they need considerable training, and hence results in longer execution times. More interestingly, the LSTM seems to be best approach for activity recognition; we've focused on the techniques that were constructed using LSTM as well as Inception. On hybridizing these two deep-learning techniques, there is indeed reliable activity recognition. This improvement was due to the lower computational complexity and higher data handling capability.

In 2018, Ma*et al.*(Ma, Chen, Kira and AlRegib, 2019) have developed and analyzed two alternative networks to better incorporate the spatiotemporal information. The two alternative networks were (1) Temporal Segment RNN and (2) Temporal-ConvNet in the Inception approach.Researchers demonstrated that on spatiotemporal feature matrices, both RNNs (with LSTMs) as well as Temporal-ConvNets had effectively leveraged spatiotemporal dynamics to substantially increase overall performance. Despite necessitating substantial temporal augmentation for end-to-end training, their empirical outcomes on the “UCF101 and HMDB51 datasets” had achieved comparable performances of 94.1 percent and 69.0 percent, respectively. Moreover, the prediction accuracy is lower (30%). Moreover, the overall computation time was lower.

In 2020, Suresh *et al.*(Suresh and Visumathi, 2020) have projected a novel “deep neural network architecture” for human action identification based on TL. Researchers have demonstrated the usage of the transfer learning to improve the characterization of human behavior with such a short video data set. Inception ResNet CNN and a LSTM were used in this research work. The model has been trained by retrieving selected features from “Inception ResNet v2”, and afterwards by applying the CNN output feature vectors towards the RNN for learning the action scene. Later, they attempted to categorize the incoming clips using the training set. The model's accuracy performance has been contrasted to that of the VGG16, ResNet152, and Inception v3 models. On the “UCI 101 and HMDB 51 data sets”, the results demonstrated that the LSTM framework employing Inception_ ResNet v2 had provided an excellent classification accuracy of 92 percent and 91 percent, respectively. Apart from this, the training loss was higher, and so the proposed model was not utilized in large-scale applications.

2.2.1 LSTM for behavioral Analysis

In 2021, Weng et al. (Weng and Jin, 2021) have suggested a new framework called the WLSTM with SME for video activity prediction. In order to employ the shortest geodesic distance in such an undifferentiated weighted graph, a “boundary-prior based motion segmentation” method was first introduced. Then, to segregate the mobile component from the heterogeneous surroundings, a dynamic contrast segmentation technique has been proposed. In each frame, the SME was further established to boost the moving objects by minimizing superfluous background. Furthermore, an appropriate long-range attention method has been intended to address the long-term interdependence of sophisticated non-periodic behaviors through dynamically concentrating mostly on semantically relevant pictures rather than interpreting all recorded frames identically. Eventually, the UT-Interaction as well as sub-JHMDB databases have been used to test the proposed approach. On the both databases, the experimental results have demonstrated that WLSTM with SME excels a variety of state-of-the-art techniques statistically. Moreover, under a low observational ratio, it was easy to misclassify activity because of the similarities between short-term actions.

In 2021, Aghaei *et al.* (Aghaei, Nazari and Moghaddam, 2021) have introduced a novel framework using “ResNet feature extractor, Conv-Attention-LSTM, BiLSTM, and fully linked layers” for activity recognition. In order to avoid the premature convergence, a sparse layer has been inserted after each LSTM layer. Optical flow, in addition to RGB pictures, has been employed to include visual movement information. Video sequences were split into equal portions based on the similarity of subsequent images. To accomplish the flow, frames from succeeding segments were employed as consecutive frames. The convolutional attention network has been used to uncover the specific surroundings. The suggested algorithm has been tested on two prominent databases: UCF-101 and HMDB-51, and its effectiveness were 95.24 and 71.62, respectively, for these datasets. Based on the data obtained, computational complexity was decreased by employing a sparse layer rather than a dropout. A deeper Network model seems to have a better recognition rate than a one-layer LSTM network. Moreover, according to the confusion matrices, error seems to occur more frequently in classes that have more similarities.

In 2019, Ma *et al.*(Ma, Chen, Kira and AlRegib, 2019) have constructed a robust “baseline two-stream ConvNet” for activity recognition. The two-stream ConvNet was built using the ResNet-101. Researchers had leveraged this as a starting place for exploring the application of RNNs and Temporal-ConvNets for retrieving spatiotemporal data. They subsequently proposed and investigated two alternative networks for significantly integrating the spatiotemporal information: "1) Temporal segment RNN and 2) Temporal-ConvNet in the style of Inception". They established on spatiotemporal features matrix, both RNNs (using LSTMs) and Temporal-ConvNets effectively leveraged spatiotemporal dynamics to substantially increase overall effectiveness. Their research highlights particular limits for each approach, which might serve as the foundation for future research. Despite necessitating substantial temporal augmentation, their empirical outcomes on the UCF101 and HMDB51 datasets had yielded better performances. The temporal information was not extracted precisely, and so the performance of the model was slight lower.

In 2021, Rao *et al.* (Venkata, Krishna Rao, Gopikrishna and SukritaLatha Parameswaran, 2021) have presented a DL model for activity recognition. It's indeed difficult to identify, recognize, or characterize human activity from a wide range of identifiers due to its complicated pattern of structuring human activity. A progress has been put in this suggested technique to distinguish the activities from big video clips and categorize them correctly using an LSTM network. On the UCF data set, features have been extracted by employing Inception V3 networks. Using this proposed method, a 96 percent HAR accuracy was achieved. Even though, the proposed work had achieved higher recognition accuracy, it isn’t applicable for large scale applications, as it has been tested with smaller sized data.

In 2021, Pandey*et al.* (Pandey and Muppalaneni, 2021) have indeed chosen clips featuring extreme classes, i.e., alert and sleepy, for their recommended research. They likewise utilized computer vision as well as a deep-learning methodology to establish two distinct models based on temporal and spatial features. The computer vision methods were used to obtain temporal characteristics, which would then be processed using LSTM, whereas the second method included a CNN to extract spatial information, which was then be processed using LSTM. Despite the fact that the temporal model seems to be more complex and also has lower accuracy than the spatial model, the research suggests a superior performance in terms of “training time than the spatial model” by comparing confusion metrics and the AUC– ROC score.

In 2019, Ahmad et al. (Ahmad, Kazmi and Ali, 2019,) have used CNN as well as LSTM to offer a unique approach for recognizing the human physical activity. Since the information collected from wearable sensors like accelerometer and gyroscope seems to be a time-series data, obtaining high accuracy using standard ML methods (such as support vector machine, K-Nearest Neighbour and random forest technique) is difficult. So, in order to achieve the maximum accuracy, researchers considered a new CNN model that consists of three CNNs that retrieve information for data collected from various sensors, and then all three CNNs were combined, accompanied by an LSTM layer as well as a dense layer. The setup of all 3 CNNs was preserved same, such that each input to CNN has generated the same number of features. They achieved higher accuracy by using suggested technique, which was comparable to traditional classification algorithms as well as other deep neural network algorithms.

2.2.2 Inception based models for behavioural Analysis

In 2019, XU*et al.* (Xu, Chai, He, Zhang and Duan, 2019.) have introduced a new approach for HAR referred as DL model (InnoHAR), by amalgamating the RNN and inception neural network. Moreover, End-to-end waveform data from multi-channel sensors was incorporated into the model. Inception-like modules extract multi-dimensional information utilizing different “kernel-based convolution layers”. Modelling for time series features seems to be possible whenever integrated with GRU, ensuring effective utilization statistical properties of images to accomplish categorization tasks. While contrasted to the state-of-the-art models, the suggested approach significantly outperformed with accuracy and strong generalization performance, as demonstrated by experimental verification on three of the most commonly used public HAR datasets.

In 2020, XIA*et al.* (Xia, Huang and Wang, 2020) have projected a deep neural network-based approach for HAR, by combining the convolutional layers and LSTM. With several parameter values, this model could automatically identify the behavioral characteristics and could classify them. The LSTM seems to be a RNN version that has been probably more suited for comprehending temporal sequences. The unprocessed information recorded via smart devices was passed into such a “two-layer LSTM” accompanied by “convolutional layers” in the developed system. Furthermore, to lessen the model parameters, a GAP had been used to supplement the FC layer after convolution. In addition, a BN had been implemented just after GAP layer to accelerate converging behavior of the solutions, with noticeable effects. Three public datasets have been used to test the model's performance (“UCI, WISDM, and OPPORTUNITY”). Ultimately, the model's overall accuracy throughout the “UCI-HAR dataset” is 95.78 percent, 95.85 percent in the WISDM dataset, and 92.63 percent in the OPPORTUNITY dataset. The outcomes indicated that the proposed model was much more consistent and capable for handling activities than most of the other earlier studies. It could not only capture behavioral characteristics in an appropriate fashion, but it also has higher complexity as well as lower precision.

In 2020, Mustafa et al. (Mustafa, Dhavale and Kuber, 2020) have presented two HAR architectures, namely “Faster RCNN Inception-v2 and YOLOv3” for efficient behavior recognition. Human activities including the "jogging, walking, and running" were deliberately addressed as activities to be identified by their postulated architecture. Researchers employed the “Faster RCNN Inception-v2 and YOLOv3” object identification algorithms that had already been pre-trained. The performance of the recommended designs was then evaluated using the benchmarked UCF-ARG video dataset. Throughout all cases, the empirical results demonstrated that perhaps the Yolov3-based HAR design outperformed Inception-v2.

2.3 Summary

AI and ML developments have resulted in excellent methodologies and tools for studying human behaviour. Owing to its broad range of applications, HAR has been one of the domains that have experienced a surge in ongoing research within the ML community. HAR one of the most useful technological instruments for assisting the elderly in their everyday lives, as well as for individuals with cognitive impairments, Parkinson's disease, alzheimer's disease, as well as. It's also beneficial in fields like transportation, robotics, and sports. DL is indeed a subfield of ML that uses sophisticated ANNs to achieve great classification accuracy in HAR. In this chapter, we investigated at some of the most recent developments in activity recognition. The LSTM and Inception models are perhaps the most popular DL models; therefore, we've spent a lot of attention on them.

CHAPTER 3

EVOLUTION OF DEEP LEARNING FOR IMAGE CLASSIFICATION

3.1 Convolutional Neural Network (CNN)

CNN short for Convolutional Neural Network (Gu, Wang, Kuen et.al ,2018) is a technique built over a neural network. The entire structure of the neural network remains the same, but an extra data processing step is added. The input variables go through a compressor function which takes in their values and spits out a singular value for the set of input parameters being convoluted.

Mostly used for training networks to recognize features in images, a set of pixel values in a localized area is given as an input to the compressor function. The compressor function can take many different parameters such as the height and width of the localized area (usually they are is taken to be a square), the step size (how many pixels are to be skipped between each input area, weights for the different positions of the pixels in the input area. All these parameters can be adjusted to not only reduce the dimensionality of the network but also help abstract out certain features from the raw data. For example, a convoluted matrix of the pixel values from an image when passed through different convolutes can help abstract the edges of objects from the image or sharpen it or blur it. This helps in detecting objects better than a regular neural network would. Although with enough data given a regular neural network can reach the accuracy, the process of convolution helps speed up the training and accuracy significantly.

The structure of CNN shown in figure 3.1 is comprised of mainly three types of layers:

1) Convolutional Layer

2) Pooling Layer

3) Fully connected layer (FC). Each layer performs some specific operation.

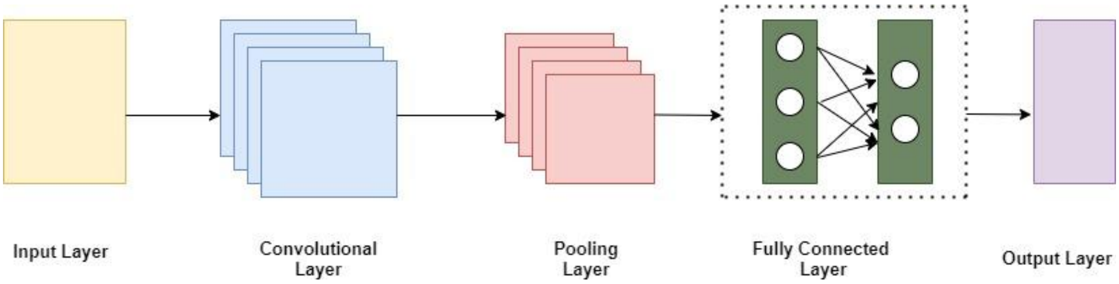


Figure 3.1: CNN Architecture

- **Input layer**

This is the first layer of CNN; it takes image as the input. Image is a collection of pixel values arranged in 3-dimensional space. A digital image consists of 3 colour channels Red, Green, and Blue (RGB). An image is arranged in 3-dimensional vector (height, width and colour channel). For e.g. If Input image has shape/dimension as 64x64x3; it means the stated image has 64 as height, 64 as width and 3 colour channels (RGB). Shape of the given input image is changed to 1-D before feeding to the model. Hence, the new shape of the data is 4096x1; the initial role of CNN is to reduce the dimension of the data, while preserving the important parameters/features.

- **Convolutional Layer**

This is the first layer after the input, which is actually responsible to extract the low-level features of an image. In this layer, mathematical operation of convolution is performed between the given input image and a filter of defined size say, MxM. The filter slides over the input image and the dot product are taken between the filter and input image as per the defined size. The output of this operation is known as feature map, which provides low level information about the image such as corners, edges. On later stage, this feature map is fed to the other layers to learn high level features of the input image. The process of convolution is shown in the figure 3.2.

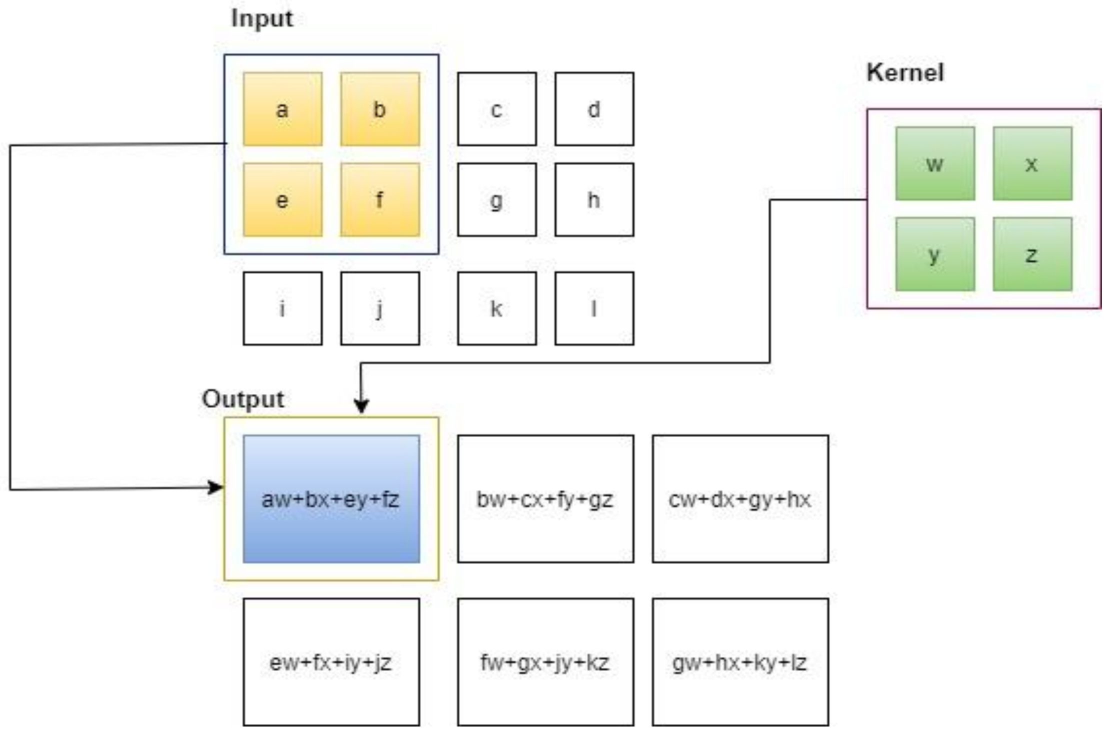
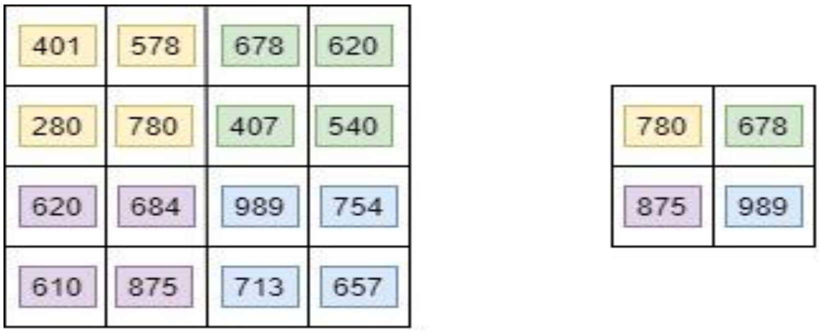


Figure 3.2: A Convolutional Operation

The filter slides over the input image with a stride S. The defined kernel/filter which can be called as weight matrix move across the entire image, moving one pixel value at a time. If the value of the stride S is set to 1, then it will move 1 pixel value. As we increase the Stride value, the size of resulting image gets decreased.

- **Pooling Layer**

The sole purpose of pooling layer is to reduce the size of convolved feature map in order to make it computational effective. Operation on each feature map is performed independently, by reducing connections between the layers. It only reduces the spatial size of the image and doesn’t impact on depth dimension of the image. Various types of pooling operations are available. Max pooling is the most common function applied on pooling layer; which selects the largest element from the feature map. Average pooling calculates the average of the elements from the predefined section of the image. Sum pooling calculates the sum of all the elements present in the predefined section. The most common used method is shown in figure 3.3.



401	578	678	620
280	780	407	540
620	684	989	754
610	875	713	657

780	678
875	989

Figure 3.3: Max pooling Operation

The resultant matrix size keeps on decreasing with each layer. This depends on the size of kernel and stride defined on each step.

- **Fully-Connected Layer (FC)**

Fully Connected layer is the last layer of any CNN model. FC layer connects each and every neuron from one layer to each and every neuron of other layer, with some weight. This layer receives the input vector and produces a new output vector. To accomplish this objective, it performs some calculations based on a specific activation function.

- **Output Layer**

This is final classification layer of the network. This layer also contains activation function which is used to calculate the probabilities of the given input to a particular output class.

3.2 Major Challenges

DL is the most powerful technique of Artificial Intelligence. It has the capability of extracting the features automatically from the given data. The performance of DL models is highly dependent on the data which is fed to the model. When employing DL model, several issues or challenges are required to be taken care of. The following issues that each DL-based model encounters are listed:

1) **Training data:** DL based model are data hungry models. They require a lot of data for the training purpose. But sometimes, it may happen that there is not enough training data. In such cases, following solutions can be adopted.

Transfer Learning: It is an approach where the model is trained and developed for some tasks and it is reused to perform other task (Weiss, Khoshgoftaar and Wang, 2016). The first model can be considered as the base for the target model. The aim of the base model is to extract the features from the given data. The required results and the parameter setting are stored as weights. These weights can be used partially or fully for another model. The weights of the base model can be transferred to some other neural network. The benefit of using transfer learning is that, model re-training is not required from scratch, which further saves the training time.

Data Augmentation: Data Augmentation (Shorten and Khoshgoftaar, 2019) helps in situations where there is limited number of samples are present in the training set. It also deals with the problem of overfitting. There are wide variety of augmentation techniques are available.

- Flipping: Flipping can be performed vertically as well as horizontally. Horizontal flip is more common as compared to vertical. The performance of the technique has been verified on ImageNet dataset. It is applicable to most of the applications but not suitable for digit recognition. Because flipping the digit ‘6’ may result in digit ‘9’ which is not acceptable. This technique helps in increasing the size of the training data but doesn’t preserve original labels.

- Color Space: Any digital image is encoded as a vector of Height x Width x Color Channel. The colored image is represented in RGB color space. To enhance the size of training data, each image can be treated as a separate channel like ‘R’, ‘G’, ‘B’ and treated as the isolation matrix by putting zeros for rest of the discarded option. Same way, other color option is selected and zeros are fed in the discarded channel.

Cropping: There is always some area in the captured frame which is of no use. Cropping helps us to slice the image in such a way that only area of interest is always focused. Figure 3.4(left) shows the centre cropping, where only the centre portion of the image is cropped. Whereas, in figure 3.4(right) shows the example of random cropping of the image.

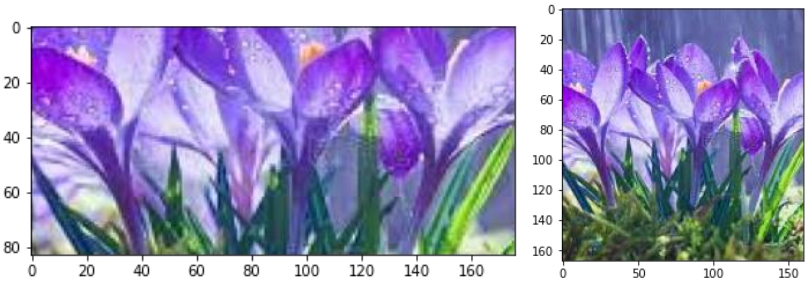


Figure 3.4: Centre cropping (Left) and random cropping (right)

- Rotation: Rotation also helps to increase the training data. The input image can be rotated along the axis and stored in the database.
- Translation: Translation augmentation is the best solution in case of images with positional variations. The input image can be transformed in up, down, left and right direction as shown in figure 3.5.
- Noise Injection: Injecting arbitrary noise deliberately helps DL models to behave more intelligently.

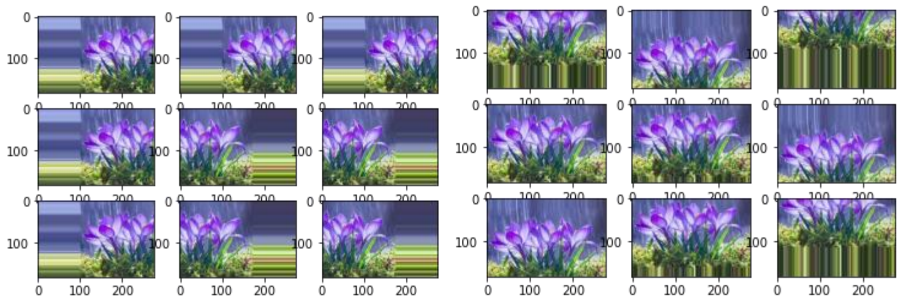


Figure 3.5: Random Images showing horizontally (left) and vertical (right) shift

2) **Imbalanced Data:** This is the biggest challenge for DL-based models. It is very difficult to have balanced data. To achieve good accuracy and results the data should not be skewed. DL based model yields low and poor accuracies

when trained with imbalanced data. Various methods are available to handle the imbalance/skewed data. The best solution is to perform resampling of the data. The objective of the resampling is to either increase the frequency of minor class (less samples) or to decrease the samples of major class (more samples). A careful measure can be opted while choosing the evaluation matrices. Area Under Curve (AUC) (Wang, Sun and Xu, 2015) can be considered for calculating the resultant loss. Also, weighted cross-entropy loss can be considered over cross entropy loss.

3) **Overfitting:** Overfitting is also a very challenging problem for DL based models. This scenario occurs when the DL model can’t generalize or fit well on unseen data. This situation leads to the degradation in network performance. The reason for this behaviour is the lack of training data. Whenever a network tries to learn from the small sample data it focuses to memorize all the datapoints instead of generalizing it. Following methods can be opted to avoid overfitting situation (Xu, Zhang, Gu and Pan, 2019):

- ***Reduce Network Complexity:*** The data simplification technique can be opted for handling the situation of overfitting. It occurs due to the complex nature of the network. Further, it is suggested to reduce the size of the model by removing some layers. Also, the complexity further can be reduced by decreasing the number of neurons in the fully-connected layer.

- ***Data Augmentation:*** The problem occurs when there is a small size of dataset. Real-world data collection can be tedious process for some domains. By applying different augmentation techniques like rotation, flipping, cropping, translation etc. the size of the existing training dataset can be increased. Augmentation technique help the model to learn from the different instances from different perspective.

- ***Weight Regularization:*** Behaviour of the network becomes abrupt, when it became overfitted. Weight regularization is the technique which tries to stabilize the network by penalizing the larger values of weights. It forces the optimizer to reduce the weights so that a better performance can be achieved.

- ***Early stopping:*** In training a neural network, with each iteration, weight of the network is updated with the aim to minimize the error. Sometimes the situation of overfitting happens whenever network undergoes for a long training process. Early stopping is a technique which gives the knowledge as with how many iterations network begins to reach overfitting situation.

4) **Vanishing Gradient Problem:** This is the most common problem for any gradient based learning technique (Alzubaidi, Zhang, Humaidi and Farhan, 2021). Gradients/derivative of the network can be found using backpropagation. Derivatives calculation can be computed by the chain rule (from final layer to the first layer). The derivatives of each layer are multiplied down the network to compute the derivatives of initial layers. Sometimes, due to small value of gradients, weight is not updated. Gradients which are almost in vanishing stage are the barrier of network training. No further training and no updates in weights leads to the situation where network will stop completely. Training with small gradients is possible only when there is a very a smaller number of layers in the network. Otherwise, with complex network with more hidden layers, value of gradient will reach almost to zero.

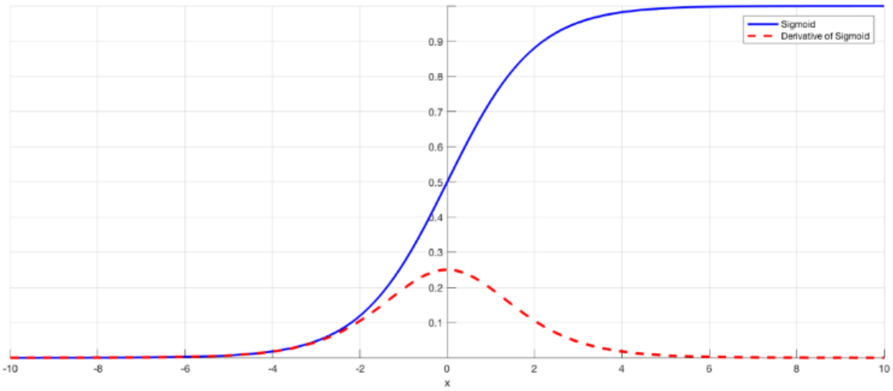


Figure 3.6: The Sigmoid Function and its derivative

Activation function (Sharma, Sharma and Athaiya, 2017) like sigmoid converges the large input space to smaller space. Therefore, a large change in the input makes a very small change in the output. Hence, the derivative of the sigmoid function becomes smaller. From figure 3.6, it is observed that increase in the input results in the decrease of the derivate. Small gradient value will not be able to update the weight of the network. Hence, other activation functions like **Rectified Linear Unit (ReLU)** can be opted. ReLU is linear (identity) for all the positive values and zero for all the negative values. It overcomes the problem of vanishing gradient and allow the model to learn faster and perform better.

Batch normalization can also be used to resolve the issue of vanishing gradient. The problem occurs when large input space is converged to smaller space, leads the derivatives to disappear. This technique can be applied on the input value which will in turn maintain the boundaries and derivatives will not go out of boundary.

5) **Exploding Gradient Problem:** Exploding gradient (Alzubaidi, Zhang, Humaidi and Farhan, 2021). behaves opposite to the vanishing gradient problem. In the former case, gradients become so small and not able to participate in the weight update of the network. Whereas, exploding gradient problem occurs when large size gradients are sent back during the backpropagation step. It will result the system into an unsteady state. Also, large gradients decrease the accuracy of learning. Various weight regularization techniques can be opted for avoiding this problem.

3.3 CNN Architectures for Image Classification

3.3.1 Pre-Trained Models

CNN architectures work on the same principle of applying convolutional operations to each successive layers to the input. Initial spatial input dimension keeps on reducing periodically whereas there is an increase in feature maps.

Traditional architectures focus on stacking of convolutional layers whereas modern architectures explore the new dimension of improving learning process.

LeNet

This is the basic CNN architecture introduced in 1998. The structure of LeNet (LeCun, Bottou, Bengio and Haffner, 1998) shown in figure 3.7, has 7 layers. The first layer considered as input layer, which takes input with dimension 32x32. Input image is convolved with 6 filters of 5x5. The result of convolution comes to be 28x28x6. The second layer is the pooling layer, which performs pooling operation with a filter of size 2x2 and stride 2. Hence, the dimension of the resulting image gets 14x14x6. In the same way, third layer with 16 filters of 5x5 and next pooling layer with a size of filter as 2x2 and a stride value of 2. The resultant image's shape is reduced to 5x5x16. Next, fifth fully connected layer with 120 filters of size 5x5. Next layer is also fully connected layer with 84 filters. Finally, the last and output layer is the SoftMax layer. This layer is used for the classification purpose.

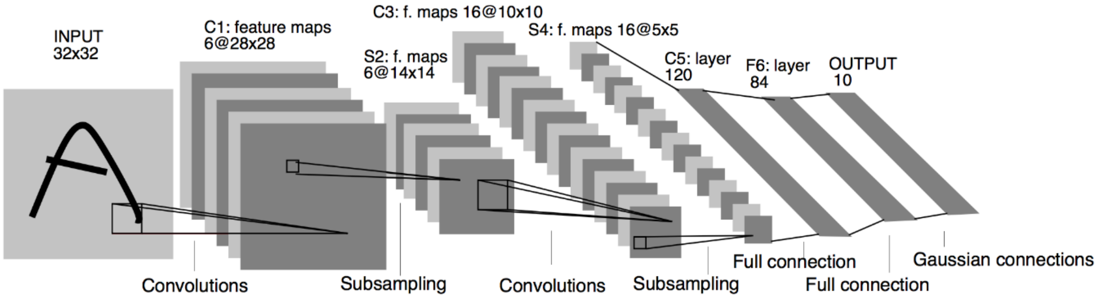


Figure 3.7: LeNet

AlexNet

Alexnet (Krizhevsky, Sutskever and Hinton, 2012) was designed by Alex Krizhevsky in 2012. It was a major breakthrough in CNN development. Innovations were added through training on multiple GPU's and using augmented techniques like flipping of the image around the axis, rotation, cropping etc. Model used ReLU activation function instead of tanh. It helped in reducing the training time as well as help in handling vanishing gradient problem

The architecture of AlexNet shown in figure 3.8, consists of 8 layers. It has 5 convolutional and 3 fully connected layers. The Alexnet used ReLU activation function after each layer. Max pooling and dropouts were also used to avoid overfitting of the data. This model was trained on the ImageNet dataset and Stochastic Descent Gradient (SGD) is used as the optimizer.

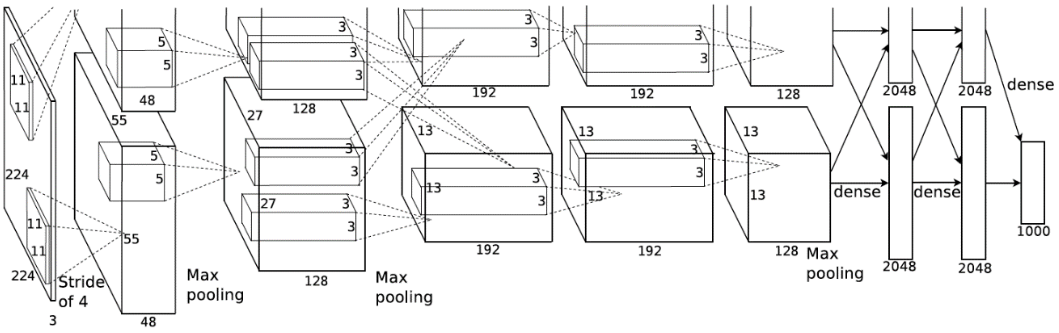


Figure 3.8: AlexNet Architecture

ZFNet

ZfNet Zeilar and Fergus (Zeiler and Fergus, 2014) presented a new CNN called as ZfNet, shown in figure 3.9. Author found that the learning mechanism of earlier CNN was unable to classify complex data. They proposed a multilayer Deconvolutional Neural Network (DeconvNet), famous as ZfNet. The idea of feature visualization was experimented and validated on AlexNet using DeconvNet. On experiment, authors found that only few neurons were active in first and second layer and some of the neurons are dead/inactive. Based on the mentioned finding, author adjusted CNN topology and performed parameter optimization. They maximized the learning of the network by reducing the filter size and strides value.

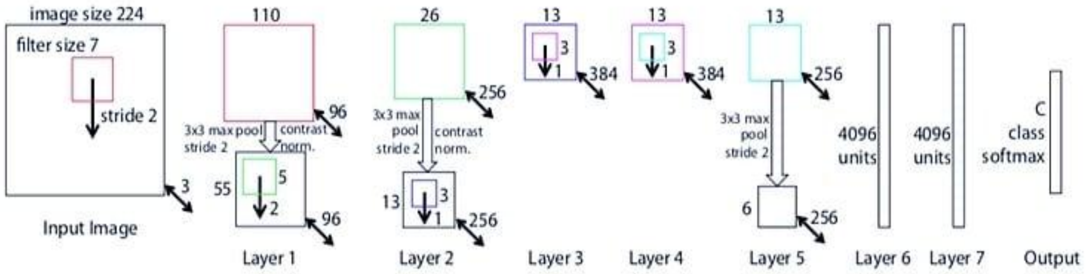


Figure 3.9: Architecture of ZfNet

VGG16

VGG (Simonyan and Zisserman, 2014) is also a very popular network architecture. It is also a CNN based architecture. VGG showed a great improvement in comparison to AlexNet. The input to the network is a fixed RGB image of shape 224x224. The image is passed through various convolutional layers with very small receptive field. In a few configurations the filter size is set to (1 × 1), which can be identified as a linear transformation of the input channels (followed by non-linearity). It replaced the large kernel size (used in AlexNet) with multiple 3x3 kernel size filters with stride of 1 in convolutional layer and uses SAME padding in pooling layers. The architecture of VGG16, shown in figure 3.10, has five max-pooling layers, convolutional layers. Max-pooling is performed over a 2 × 2pixel window, with stride 2. A stack of convolutional layers is followed by three Fully-Connected (FC) layers. The idea of using multiple non-linear stacking of smaller size filters over the one with larger size enhances the depth of the network also enables the model to learn more complex features at a reduced cost.

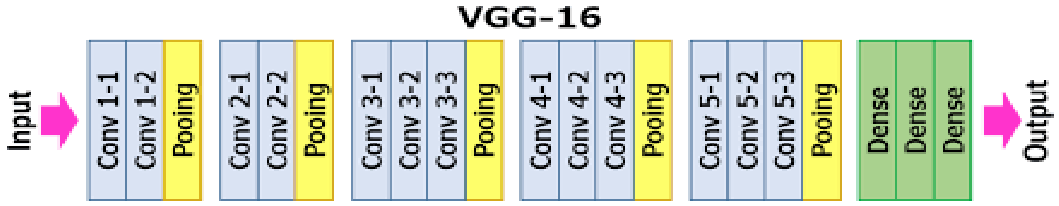


Figure 3.10: Architecture of VGGNet (configuration D, VGG16)

Inception v3 Network

The inception network was introduced in 2014. The first version of inception was popularized as GoogleNet (Szegedy, et.al, 2015). Later on, the model was modified and improved by adding batch normalization, hence inception v2 was released. Further, the architecture’s modalities were enhanced to **factorization convolution**, to modify **auxiliary classifier** and introduced the **grid size reduction**. And this modified version was released with name inception v3.

The idea of using Factorization Convolution is to decrease the count of parameters without decreasing the efficiency of the network. The technique used in Inception v3 is as follows:

- **Factors into smaller convolutions**

The aim of making factors into smaller convolution is to increase the number of convolutional layers by stacking them in depth as shown in the figure 3.11. The kernel size is reduced at every layer, which makes it more capable of extracting varied and complex features of the image, passed to it. For example, by using 7x7 kernel filter dimension in 1 layer has 49 parameters while using 2 layers of 3x3 filter dimension results in (3x3+3x3) =18 parameters. The number of parameters is reduced to nearly 63%, a significant drop can be seen in number of parameters.

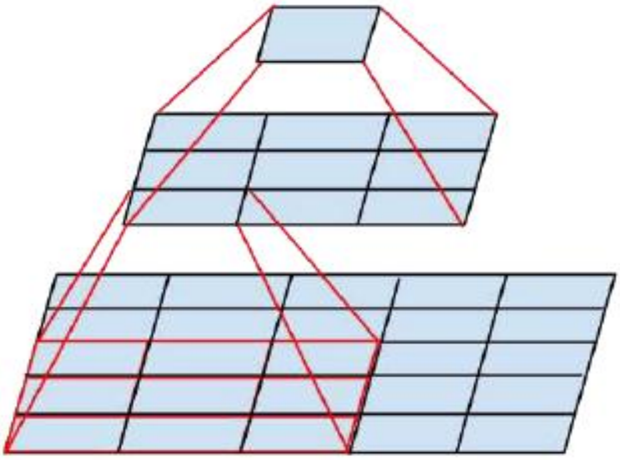


Figure 3.11: Graphical representation of factors into smaller convolutions

- **Factorization into asymmetric convolutions**

The purpose of making asymmetric convolutions is to reduce the parameters by adding some asymmetry to the network. The idea behind this method is to replace NxN filter with 2 consecutive layers with 1xN and Nx1 size. For example, by using 5x5 filter in layer, results in 25 parameters while using 1x5 and 5x1 the number of parameters dropped to 10. Hence significant drop in parameters is observed. Figure 3.12 shows that replacement of 3x3 convolution kernel with 3x1 followed by 1x3.

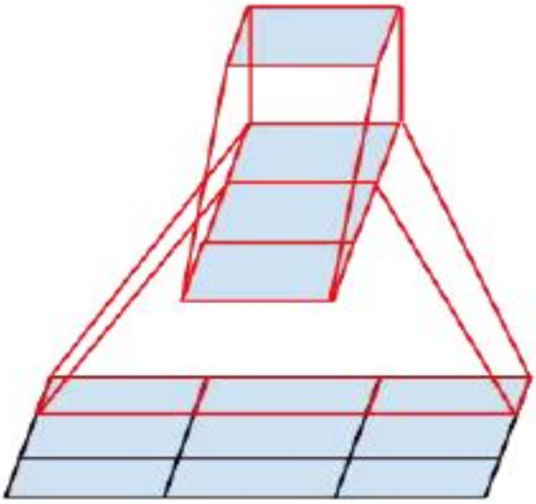


Figure 3.12: Graphical representation of Factorization into Asymmetric Convolutions

- **Auxiliary Classifier**

Auxiliary Classifiers (AC) are used to improve the convergence of deep networks. They are usually attached before the final layer of the network. The idea of using it is to handle the vanishing gradient problem. The first version of inception network used 2 auxiliary classifiers. It is inserted between layers during training process and the loss incurred is added to the final loss of the network. Thereafter, modifications were performed in the later version and 1 auxiliary classifier is used in Inception v3. The purpose of using the Auxiliary Classifier was also different: initially it was used to allow having a deeper network, with the v3 it acts as a regularizer to the network.

- **Grid Size Reduction**

To reduce the number of weights, max pooling layer is added before the convolutional layer. But sometimes this is not the efficient solution. Adding pooling layer after the convolutional layer seems to be costly affair. The Grid size reduction provides the efficient solution by reducing the parameters and provides a solution where in a layer there is both part of convolutional and max pooling is concatenated**.** Figure 3.13 shows the structure of Inception v3 network.

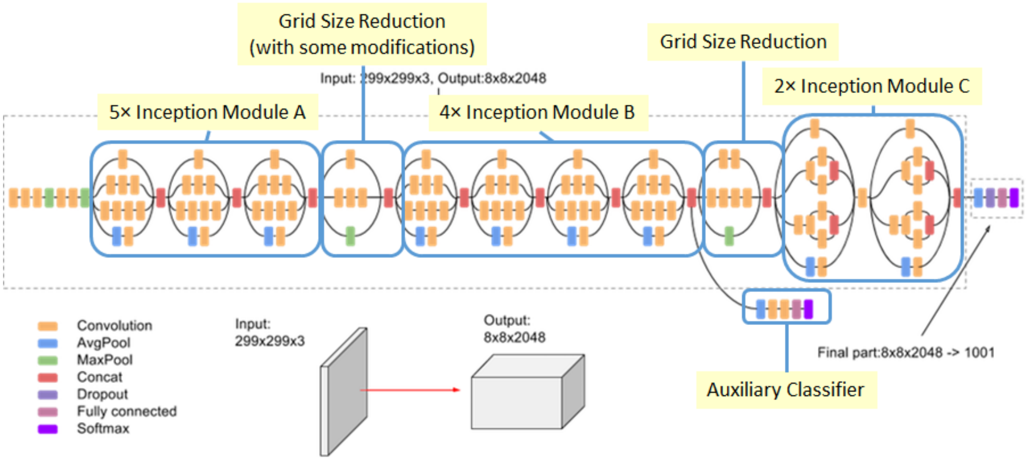


Figure 3.13: Inception-v3 Network Architecture

The network is divided into modules A, B and C depicted in figure 3.13

Module A

Module A of inception network supports the factorization into smaller convolutions. It replaces 5x5 Convolutional layer with two 3x3 ones as shown in the figure 3.14.

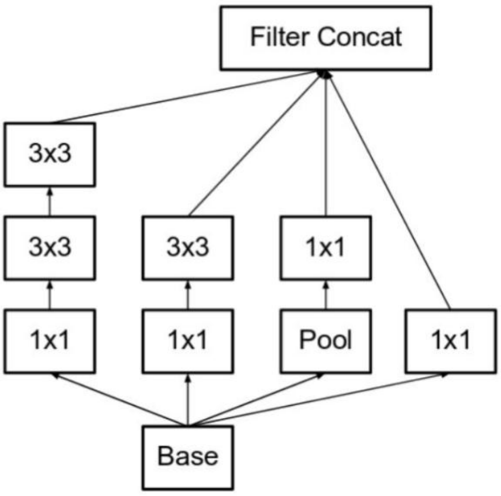


Figure 3.14: Inception Module A

Module B

It refers to factorization into asymmetric convolutions. It takes nxn convolution and replaces it with 1xn convolution followed by nx1 convolution as shown in the figure 3.15.

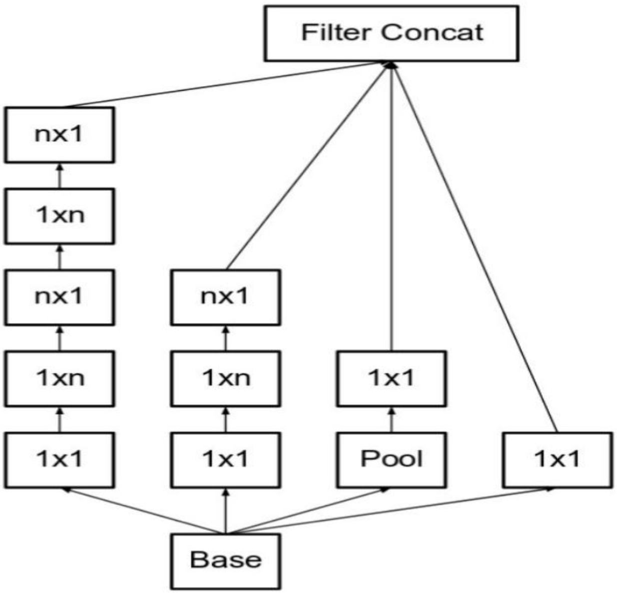


Figure 3.15: Inception Module B

And

Module C represents the combination of tiny and asymmetric convolutions. It takes 3x3 output of the inception module and expand it by replacing with 1x3 and 3x1 convolutions in parallel way as shown in figure 3.16.

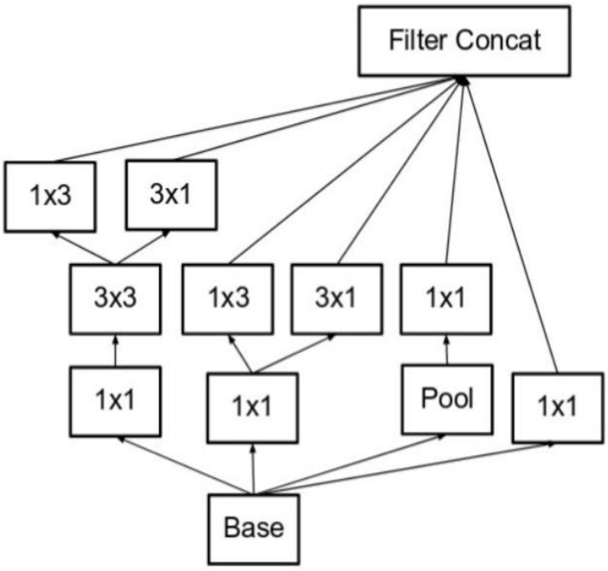


Figure 3.16: Inception Module C

ResNet

ResNet is a family of CNN architectures. It has 18 to 152 deep convolutional layers. Since, the deep neural architecture suffers from vanishing gradient problem. He et al. proposed a new architecture ResNet (He,Zhang, Ren and Sun, 2016.). This architecture introduces the use of “Residual block”. In this network, a technique of skip connections is used to solve the problem of vanishing gradient. The skip connection skips the training from few layers and connects directly to the output. The advantage of using skip connection is, if any layer degrades the performance of the network, then it is skipped by performing regularization. It uses residual mapping (1) instead of learning a direct mapping (2). Figure 3.17 shows the mapping in residual block.

$$H(x) = F(x) + x \tag{3.1}$$

$$H(x) = F(x) \tag{3.2}$$

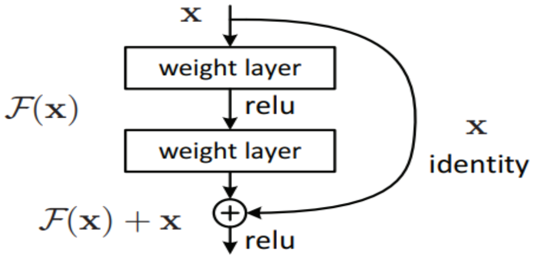

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 3.17: Residual block

3.4 Evaluation Criteria for CNN Models

The CNN models are evaluated on the basis of following parameters:

1) **Accuracy:** It is important for any project to give the better accuracy as compared with literature. Various models have been discussed above for the image classification. Inception v-3 demonstrates that is the best network among the stated networks in terms of accuracy by achieving a top-error of 3.5% in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Szegedy, Vanhoucke, Ioffe, Shlens and Wojna, 2016).

2) **Network Size**: The size of the network depends on the number of layers present in it. (Szegedy et.al, 2016). This selection criteria are beneficial at the point of determining other aspects related to computational or time complexity. The aim of Inception network is to achieve high accuracy and reduced computational cost. The architecture contains a total of 48 layers. It uses network branches instead of sequential increment, in its architecture. Merging and splitting is the key idea behind the feature extraction process.

3) **Computational Complexity**: It is important to estimate the computational complexity of a network before it is trained on specific computer. The complexity of the network dominates with the increase in number of convolutional layers. It is suggested to estimate number of Floating Points Operations Per Second (FLOPs) and parameters of the chosen network for the estimation of computational complexity.

3.5 Analysis of various CNN Architectures

Table 3.1 gives the analysis of structural details of various architectures discussed above. The complexity of the network also increases with the increase in number of convolutional layers. With the increase in convolutional layer, there is a considerable increase in the number of parameters in each layer. Table 3.1 provides a detailed analysis of various structures in terms of input shape, dimension explored and parameters used. Also gives insight to the error rate for each architecture.

Table 3.1: CNNs Architecture's analysis

Year of Proposed Work	Model	Depth	Error rate	Input Size	Parameters Used	Dataset used	Dimension Explored
2012	AlexNet	8	16.4	227x227x3	60 M	ImageNet	Spatial
2014	ZfNet	8	11.7	224x224x3	60 M	ImageNet	Spatial
2014	VGG	16,19	7.3	224x224x3	138 M	ImageNet	Spatial
2015	GoogleNet	22	6.7	224x224x3	4 M	ImageNet	Spatial
2015	Inception v3	48	3.5	229x229x3	23.6 M	ImageNet	Depth and Width
2016	ResNet	152	3.57	224x224x3	25.6 M	ImageNet	Depth and multi-path

3.6 Summary

In this chapter, focus is on understanding the architecture of CNN. The layers arrangement and operation performed in each layer is understood. There are various challenges associated with DL models, are discussed and few techniques are also presented in order to handle mentioned constraints. Further, the advancement in CNN architectures is understood. Also, focused on the evaluation criteria for the CNN models and detailed structural analysis of various CNN architecture is performed and complied.

CHAPTER 4

INCEPTION AND LSTM BASED METHODOLOGY FOR HUMAN ACTIVITY RECOGNITION

4.1 Background

Human activities can be recognized according to the nature of input data. It can be recognized either through images or through videos. Former method involves of identifying actions like sitting, eating etc. This recognition is possible with static images. Some activities like robbery, hitting, running, fighting etc. cannot be identified with just a single frame. There is a need to have some extra information related to prior and post occurring of the event. Examining the previous and next frame in correspondence to selected frame is important to identify and to establish the relationship between successive frames.

Videos are the collection of image frames. Image conveys the visual information at its own. From the given image, one can easily analyze the activity or event that is happening. In today’s world, applications like HAR in case of surveillance, health monitoring system, day care service etc. use the powerful medium to capture the ongoing activity. The captured videos are processed to analyze the situation so that further action or alert can be generated. To understand the need of the system and to enhance the visualization of the image, it is necessary to preprocess the data for further analysis. Removal of unwanted noise is one of the basic steps of digital image processing. Noise gets introduced at the time of data collection. Usually, image gets faulty at the time of acquisition and transmission. Camera quality, distance from the camera, occluded background are some of the reasons to enhance the image quality. Removal of noise is necessary to get clear image. Also, aim to retain the important parameters of image for proper classification.

In literature, human activities are analyzed through traditional handcrafted feature extraction methods (Zhu, Shao, Xie and Fang, 2016). With the advancement of technology, methods based on DL is getting so much attention. These methods have the capability to learn the features automatically. There are variety of methods; CNN is one of them, which is comprehensively explained in sub-section 3.1 and other important method for handing temporal structure, Long Short-Term Memory (LSTM) is discussed further.

Recurrent Neural Network – Long Short-Term Memory (LSTM)

LSTM is powerful and well-known type of recurrent neural network which is capable of learning patterns and most suitable for sequence prediction problems. Architecture of LSTM was proposed in (Hochreiter and Schmidhuber, 1997). LSTM network has feedback connection therefore well suited to learn from experience. It cannot only work for single data points rather it works on the complete sequence. Figure 4.1 shows the basic LSTM cell. The LSTM cell has a linear unit with self-recurrent connection, input, output and forget gate. The self-recurrent connection (Staudemeyer and Morris, 2019) has a weight of 1.0 and it guarantees that the state of the cell remains stable from one time step to other. Also ensures that no outside interference is permitted. Other gates present keeps track of the in and out flow of the information. Forget gate keeps check on the cell state. The information which is no longer is required by the cell is removed by the forget gate. LSTM network is most suitable for time series data. Activity Recognition depends upon the previous and next frame chosen as per the selected key frame. Analysis of sequential frames are required to recognize the activity correctly.

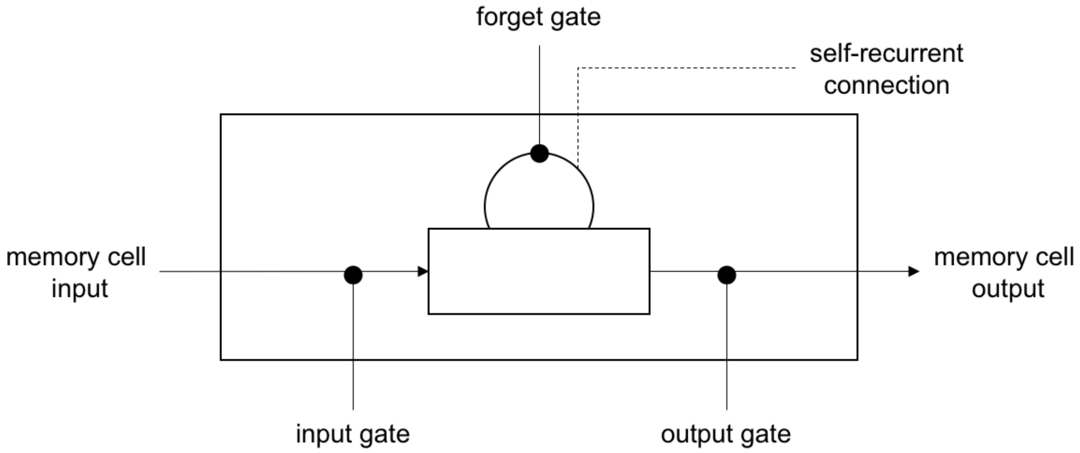


Figure 4.1: A LSTM Memory Cell

Each cell's and gate's state is represented by multiplicative equations that are:

$$\mathrm{i_t = \sigma(W_{xi}\, x_t + W_{hi}\, h_{t-1} + W_{ci} c_{t-1} + b_i)} \tag{4.1}$$

$$\mathrm{f_t = \sigma(W_{xf}\, x_t + W_{hf}\, h_{t-1} + W_{cf} c_{t-1} + b_f)} \tag{4.2}$$

$$\mathrm{o_t = \sigma(W_{xo}\, x_t + W_{ho}\, h_{t-1} + W_{co} c_{t-1} + b_o)} \tag{4.3}$$

$$\mathrm{c_t = (f_t\, c_{t-1} + i_t tanh(w_{xc}\, x_t + w_{hc}\, h_{t-1} + b_c))} \tag{4.4}$$

$$\mathrm{h_t = o_t tanhc_t} \tag{4.5}$$

W is the weight matrix, x is the input, σ is sigmoid and tanh is the hyperbolic tangent activation function. The letter i,f,o used for the respective gates and c is used for memory cell.

Convolutional Neural Network (ConvNet/CNN)

A CNN is a type of Artificial Neural Network (Ha, Yun and Choi, 2015) that is comprised of neurons and have ability to self-optimizing through learning procedure. CNNs have showed remarkable performance in the field of video classification (Ye, Wu, Zhao, Wang, Jiang and Xue, 2015), object detection (Ren, S., He, K., Girshick, R. and Sun, J., 2015), face recognition (Hu, Yang,Yi, Kittler,Christmas, Li and Hospedales, T., 2015), pattern analysis, etc. The CNN requires much lesser pre-processing operations as compared to traditional handcrafted feature extraction methods (Georgiou, Liu, Chen and Lew, 2020). Traditional methods require a lot of human intervention while CNN has the ability to learn and optimise by itself. The structure of CNN is inspired by the visual cortex (Schmidhuber, 2015) and connectivity pattern of neurons present in the human brain.

Initial spatial input dimension keeps on reducing periodically whereas there is an increase in feature maps. Traditional architectures focus on stacking of convolutional layers whereas modern architectures explore the new dimension of improving learning process. Various CNN based architectures (Alzubaidi et.al, 2021) are available for extracting the high-level features from the image. LeNet, AlexNet, VGG, Inception, ResNet etc are widely used pre-trained models for the image classification.

Inception network is used for extracting spatial features from the image. The model is chosen for the reason because it uses multiple convolutional filters like 1x1, 5x5, 7x7 which allows model for multi-level feature extraction. Multiple features extracted through multiple filters, improves the performance of the network model.

4.2 Motivation

Now a days, with the increasing need of personal safety and security, there is a need of video surveillance. The consequence of this need has led to deployment of video cameras. The motivation behind the proposed work is to focus to identify unusual/abnormal/suspicious activities poses by human being. Suspicious human activity detection from video surveillance system is gaining more popularity in public as well as indoor places. It involves recognition of human activities and classifies them into usual/normal or unusual/abnormal activity. Normal activities are the actions which person possess in his daily routine like walking, jogging, handshaking with other people etc. whereas abnormal or unusual activities are like fight, theft, riots, attack, leaving explosive baggage etc. A human operator cannot monitor such activity for a longer time due to fatigue issue, biased nature; it could lead to the occurrence of dangerous situations. This leads to a dire need for developing an automated surveillance system which can monitor such activities automatically.

The important chore of an automated surveillance system is to capture and monitor video from the cameras mounted at different locations. Further, to identify any unusual activities happening in sensitive areas. Unusual activity can be dangerous for other person(s), so it is required to monitor these activities and recognize such involved human beings, so that threat can be avoided and if event happens necessary action can be taken.

4.3 Proposed Architecture- Incept_LSTM

The proposed deep neural network architecture Incept_LSTM is shown in the figure 4.2. Incept_LSTM is trained and validated on UCF-Crime dataset. The dataset is very huge and contains long untrimmed videos. The dataset is divided into 13 anomalous categories along with 1 normal class data (discussed under experimental settings in section 4.4). Due to the limitation of the application and storage space, not all the categories are focused on. Suspicious behaviour is extracted only from videos that exhibit a scenario of assault, fight and burglary. Normal behaviour samples are extracted from the normal class.

4.3.1 Spatial Feature Extraction

The basic requirement of DL model is the large amount of training data. Also, requires high-capacity processing units. To handle the need of the current system, pre-trained model, Inception v3 is used for extracting the hidden features. The model utilizes the concept of Transfer Learning, where the learning is transferred from the base model to the target model. Inception v3 is pre-trained on ImageNet dataset (Krizhevsky, Sutskever, and Hinton,2012), consisting of 1000 categories, used as a fundamental step in the proposed model.

Inception network is used to extract the spatial and complex features from the given data. The purpose of using inception network (Szegedy, Vanhoucke, Ioffe, Shlens and Wojna, 2016) is to enhance the performance of the network by employing various sizes convolutional filters, also it reduces the computational resource usage.

Initially, to improvise the performance of the deep networks, it is assumed and adopted to increase the depth of the network by increasing number of layers and also the width of the network is also enlarged by increasing the number of nodes in each layer. This approach is easy and adopted for training large quality models. However, this simple approach has few limitations. Firstly, larger network size means more parameters and hence this expanded network is more prone to overfitting. Secondly, as the network grow exponentially, it will demand high computational resources.

Inception network is emerged as a solution. It has lesser number of parameters as compared to AlexNet. This architecture is not just computationally convenient, also provides the best recognition accuracy. Inception network got recognition after winning the ILSVRC 2014 (He, Zhang, Ren and Sun, 2016) competition for image classification. In terms of memory and parameters, initial version which was famous as GoogleNet (Szegedy, et.al, 2015) uses 4M parameters, which are very less compared to the parameters used in AlexNet (Krizhevsky, Sutskever and Hinton, 2012). Improvements were made in later versions where the model is scaled computationally with more factorization and regularization.

Multiple features are extracted using multiple filters of varying size. It uses filters of various sizes like 1x1, 3x3 and 5x5. Filter of 1x1 size is used to activate the combination of multi-channel. The collected information is passed to the next upper layer. Multiple features extracted through multiple filters improves the performance of the network model.

4.3.2 Temporal Feature Extraction

Temporal information plays an important role in understanding the complex actions. Simple actions like standing and sitting can easily be identified using a single frame. But complex activities like running, falling, fighting etc. need more temporal information for the correct classification.

To obtain the temporal information, Recurrent Neural Network (RNN) especially Long Short-Term Memory (LSTM) (Murad & Pyun,2017) is majorly adopted for learning long sequence tasks. LSTM uses a mechanism based on gates, which perform component wise multiplication of the input. LSTM block has three gates: input, output and forget gates. These gates perform the function of write, read and reset for each cell. Cell state is the key component of transferring the information from one LSTM block to other.

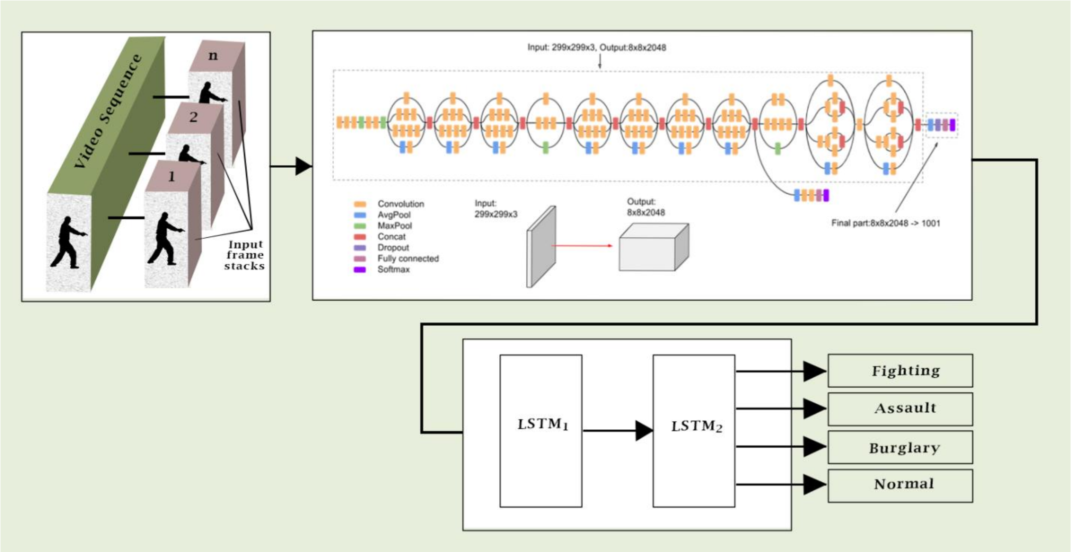


Figure 4.2: Proposed Approach Incept_LSTM

Multilayer LSTM

The efficacy of deep neural network is dependent on the number of layers attached in the architecture. To improve the learning ability and the functionality of RNN is extended. In Incept_LSTM, similar approach is followed by stacking the network with two LSTMs. This stacking of two LSTMs results in learning high-level temporal and sequential information. In traditional LSTM, data is transferred to a single layer before generating the output. However, in the applications like HAR, where more temporal information is required to understand the activity, emphasis to get more information across multiple layers. To obtain more temporal information, stacking of LSTM is performed. Each layer in LSTM network, accepts the hidden state of the preceding layer as input and computed output is transferred to the next layer. The final output is computed through Softmax classifier in the prediction state of the model.

4.4 Experimental Settings and Initial Preparation

Settings in training phase

For the suspicious behaviour analysis, multi-class UCF-Crime dataset is used. The dataset consists of 1900 real-world surveillance videos. The dataset is divided into 14 classes: Normal, Fighting, Explosion, Abuse, Assault, Road Accident, Arrest, Burglary, Robbery, Arson, Shooting, Stealing, Shoplifting, and Vandalism. Figure 4.2

shows the video frames belonging to each class. The present 1 class belongs to normal category and 13 classes in the dataset belongs to the anomaly/suspicious class. The dataset is solely focus on normal vs anomalous categories. Given video classification is 128 hours long. If all the dataset is downloaded and used for the experiment purpose then it would have taken more than 100GB of storage space. So, to avoid the complexity, I focused on only 4 classes out of 14. The chosen categories for the classification are: Normal, Fight, Assault and Burglary.

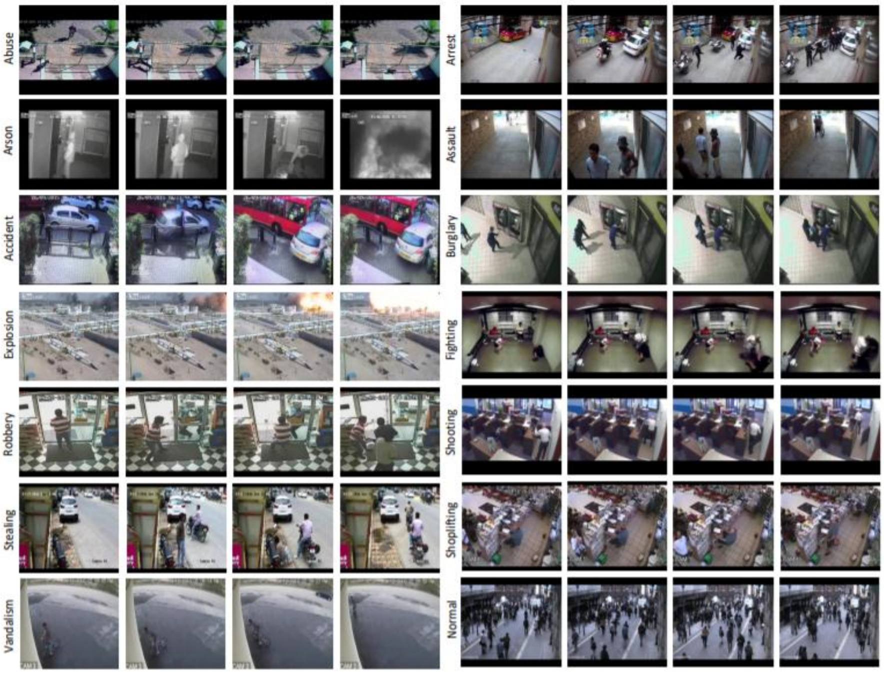


Figure 4.3: UCF-Crime Dataset Categories

Segmenting the Data

In case of Video files, Segmentation is the important step for the data preparation. Once the data is prepared, it is fed to the model for training purpose. Due to the limitation of the application and storage space, not all the categories are focused on. Suspicious behaviour is extracted only from videos that exhibit a scenario of assault, fight and burglary. Normal behaviour samples are extracted from the normal class. It is not necessary that complete video file is containing the abnormal activity. So, this is the need of system to segment the video in such a way that the segmented portion contain the required class activity. Video sample segmentation doesn't require any specific methodology for further analysis. In some case, manual segmentation is performed and, in few cases, some random samples are taken for the analysis.

Frame Generation

Frames are extracted from the segmented video files for further processing. 500 images per category is considered for training *Incept_LSTM.*16 frames per image are extracted resulting in 500x16=8000 frames per category. After getting a video sequence, frames have been extracted and those 16 frames are stored in a lexicographic order (continuous sequence) so that the temporal sequence could be recreated. Now, each image is consisting of 16 frames and size of each frame is 128x128. Therefore, size of the one image comes out to be 2048x128.

Pre-processing

Before the data is fed into the model, it is required to pre-process it. Multiple pre-processing steps are performed on the raw data to obtain fine and clean information. Most of the times, when videos are taken from the CCTVs, have noise present in it. Performing classification on noisy and corrupted images may result into incorrect classification. Hence, it is required to clean the data before proceeding further.

Data Augmentation

The performance of DL networks improves with the increase amount of data. Data augmentation is a technique that aims to increase the number of samples present in the dataset. Domain-specific techniques (Ilse, Tomczak and Forré, 2021) are applied to increase the size of training set. Flipping, rotation, crop, zoom and brightness are the common augmentation techniques used for image classification. Not all the techniques are advised to apply on the data. As vertical flip is not worth to perform on videos frames. It will not be seen as upside down. For this work, resize, horizontal flip, rotation and normalize techniques are used.

4.5 Performance Measure

It is important to measure the performance of the trained model to verify its behaviour on unseen data. Various methods have been adopted to measure the performance of the model (Minnen, Westeyn, Starner, Ward, and Lukowicz, 2006) are discussed:

1. **Confusion Matrix**: A confusion matrix represents the predicted results and describes the performance of the classification model. Each prediction can be one of the four outcomes:

True Positive (TP): When the prediction is true and the actual is also true.

True Negative (TN): When the prediction is false and actual is also false.

False Positive (FP): When prediction is true but false in actual.

False Negative (FN): When prediction is false but true in actual.

Accuracy of the model can be computed by taking the average of values lying across the main diagonal. It can be computed as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{4.6}$$

2. **F-Measure or F1-Score**: It is used to measure the test accuracy. It is the harmonic mean of precision and recall. The value of result lies in between 0 and 1. If the chosen dataset is imbalance, then classifier will not give the accurate results. And if each instance is classified as the majority class and overall classification accuracy is used to evaluate the performance of the model, then it is not a good choice for performance measure. Whereas, F-measure considers both false positive and false negative for evaluation. It is computed as:

$$F1 = 2 * \frac{precision*recall}{precision+recall} \tag{4.7}$$

Whereas, precision is fraction of correctly identified instances among all the predicted positive cases. It is computed as:

$$Precision = \frac{TP}{TP+FP} \tag{4.8}$$

Recall is a fraction of number of correct positive results over the number of all the relevant samples. It is computed as:

$$Recall = \frac{TP}{TP+FN} \tag{4.9}$$

F1-score is a more useful and accurate performance measure/indicator than accuracy. It is a good and accurate measure when the dataset is imbalance.

4.6 Experimental Results and Analysis

Incept_LSTM, model is trained in fully supervised manner. Each layer present in the architecture has some weight and bias which is initialized by some random values. Gradient is computed and back propagated. Cross entropy is used as a loss function. It is used to measure the difference between real and probability distribution for a given random variable. In this work, cross entropy loss function is used to measure the error between a true and a predicted value. Aim is to minimize the loss function, therefore, during the training process, the value of weight/parameter is also updated. Stochastic Gradient Descent (SGD) is selected as an optimizer. It is a stochastic optimization algorithm in which weights are updated for each training sample or a small subset of data. For the training phase, batch size is set to 4 and epochs are set to 50. Furthermore, a small learning rate of 0.001 is set to enhance the learning and fitting ability. Table 4.1 provides the computed results for proposed approach.

Table 4.1: Incept_LSTM Results

Dataset	Epochs	Learning rate	Optimizer	Max. Validation accuracy (%)	Min. Validation Loss (%)	Activation Function	No. of output classes
UCF Crime dataset	50	0.001	SGD	91	0.33	Softmax	4

Accuracy comparison is made in Table 4.2. The proposed approach performs well when compared with the literature. *Incept_LSTM* is compared with the approach presented in 2017, based on ensemble of multiple LSTM. This ensemble method obtained 72% accuracy on opportunity dataset, 85% accuracy on PAMAP2 dataset and 90% on Skoda dataset. Besides, a comparison is made with the other method, proposed in 2018. This approach worked on two different streams of CNN. Input from two different streams is taken and processed further. Accuracy is computed and observed 80.5%. Hence, to improve the accuracy rate, *Incept_LSTM* is proposed and it achieved 91% accuracy on UCF-Crime dataset.

Table 4.2: Accuracy comparison of the proposed method with existing methods

Evaluation Metric	**Method 1 (LSTM Ensemble) (Ordonez et. al)**	**Method 2 (Two-stream convolutional networks (Luo et.al)**	**Proposed Method (Incept_LSTM)**
Year	2016	2018	**2020**
Accuracy	72% (opportunity dataset) 85% (PAMAP2 dataset) 92% (Skoda dataset)	80.5% (own dataset)	**91% (UCF Crime Dataset)**

The proposed *Incept_LSTM* model is evaluated on pre-existing evaluation techniques. F1-score is computed to measure the fairness and consistency of the results. It is the harmonic mean of precision and recall. The value of result lies in between 0 and 1. A comparison is made in Table 4.3, a technique based on LSTM ensemble gives a F1-score equals to 0.61 whereas *Incept_LSTM* proves better in the classification and presents F1- score to be 0.89.

Table 4.3: F1-Score Comparison of Incept_LSTM with existing work

Methods	**F1-Score**
LSTM Ensemble , 2016	0.61
Proposed Method (Incept_LSTM), 2020	**0.89**

Confusion matrix is a technique which is used to summarize the performance of the classification algorithm.

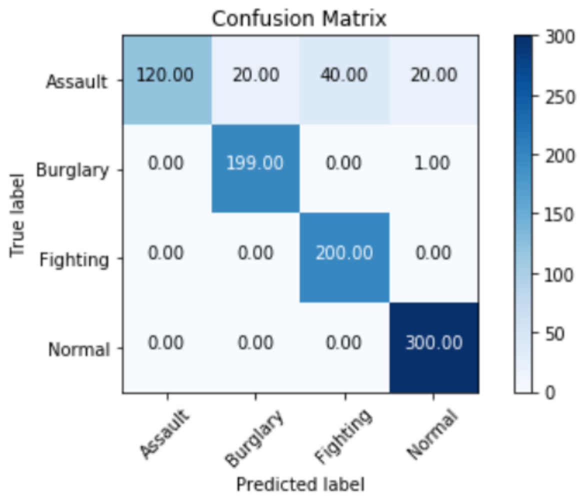


Figure 4.4: Confusion matrix of action recognition using Incept_LSTM

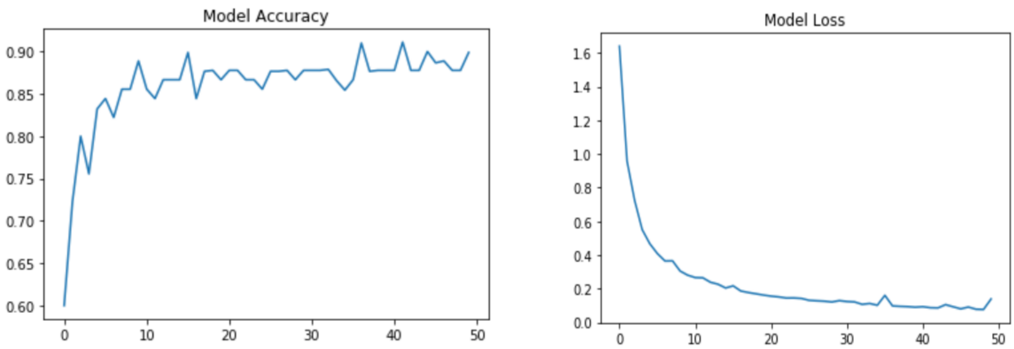


Figure 4.5: Model Accuracy and Loss

Hyperparameters Tuning and its Major Concerns

Hyperparameter setting has a great impact on model performance. Hyperparameter settings often differ for different datasets. Therefore, tuning of hyperparameters is required. Setting of hyperparameters is usually a meta-process, where each time parameters are set for the training and outcome is observed. Usually, settings are freeze with a predefined terminating criterion. This process of setting the hyperparameters to obtain optimal performance, is known as hyperparameter tuning (Yu and Zhu, 2020).

In this work, various *hyperparameters* tuned are activation function, optimizer, learning rate, batch size, and epochs. *Activation function* is used in each layer starting from input to output layer. Data is fed into input layer, followed by hidden layer and final output layer. The input values keep on changing while moving from one layer to another (as per the weighting criteria). Sigmoid, tanh, ReLU, softmax are among the variety of activation functions. In this work, Softmax activation function is used. SoftMax activation function distributes the probability throughout every node present in the output layer. This function transforms the input value into output in such a way that the total sum of the probability lies in between 0 and 1.

Optimizers have also a great impact on the training process. In the proposed work, during the training process, parameters are updated, to minimize the loss function and to attain optimal solution. SGD optimizer is used due to its property of being fast and computationally less expensive. Further, *learning rate*, is responsible for affecting the training time and accuracy of the model, so it is empirically chosen as 0.001. Selection of learning rate is a very challenging task. Small value may result in long training process whereas large value may fasten the process and leads to an unstable state.

Batch size is related with the count of training samples to work with. Usually, the size of batch should be greater than 1 and less than the training datasets. It has a great impact on accuracy of the model. Sometimes, large batch size yields in large accuracy but poses a constraint to computational cost. For training *Incept_LSTM* the batch size of 8 is chosen empirically. Besides, *epochs* are also having a large effect on the performance of the model. It is basically, the number of iterations that a model has to undergo during training process.

It is observed from the Table 4.1, empirically tested hyperparameters gives the accuracy of 91%. The performance of the model can be improved by carefully selection of hyperparameters and considering the third dimension for the video data. Chapter-5 discusses the various improvisations made in order to achieve the better results.

4.7 Summary

Understanding the current need of the system, an automated system for monitoring human activity is designed. The model utilizes the functionality of pre-trained Inception v3 for deep feature extraction. Further, use of LSTM justifies the role of sequential learning in case of activity recognition. The model is trained and validated on UCF-crime dataset with accuracy score of 91%. The resultant accuracy is compared with the other methods proposed in the literature. To measure the fairness of the result, F1-score is also computed and compared with the literature.

CHAPTER 5

A HUMAN ACTIVITY RECOGNITION MODEL BASED ON MOTION FLOW USING INCEPTION AND LSTM

5.1 Introduction

Performance of DL models is highly dependent upon various hyper-parameters (He, Zhao and Chu, 2021). As compared to traditional ML algorithms, deep neural network has more dependency on hyper-parameter tuning. Weights of neurons are initialized and updated through learning process. Whereas, other hyper-parameters cannot be estimated through any data learning process. These parameters must be set before the training proceed. Hyper-parameters are the parameters that are used to configure the DL model with or specify the algorithm used to minimize the loss function. Optimizers and activation functions plays a significant role in minimizing the loss.

To build an optimized model, various possible range of solutions are explored. The process of determining the right combination hyper-parameters that allows model to give maximum performance, is known as hyper-parameter tuning.

DL models have higher dependency on hyper-parameter tuning as compared to traditional ML models. Because DL models have more hyper-parameters to tune and also the performance depends on the configuration of hyper-parameters. Literature reports that the DL model accuracy fluctuates from 30-90% due to different selection of hyper-parameters.

5.2 Motivation

There are two types of parameters in DL models. First, *model parameters* which are learned during the training process and second, *hyperparameters* which are adjustable in nature. In this chapter, an empirical approach is used to optimize *Incept_LSTM* (reference Chapter 4) by adjusting the values of these hyperparameters. The error (after minimizing loss) is computed and used for bias-variance adjustment.

5.3 Proposed motion flow based Incept_LSTM architecture

Deep convolutional model outperforms the traditional approach of feature extraction. However, it has certain limitation (Pareek and Thakkar, 2021) like 1) deep models are generally designed by trial-and-error process, which requires large amount of labelled training data. 2) Also, large number of neuron connections will result into large computational expense.

In this section, *Incept_LSTM* (Girdhar, Johri and Virmani, 2020) is improved and optimised to analyse the abnormal and normal behavior in real world scenario. The proposed *motion flow based Incept_LSTM,* fetches the spatial information present in the RGB frames and also collects the motion information from given input data. Appearance features (Dubey, Boragule,Gwak and Jeon, 2021) are contained in RGB frames, which are manually extracted under pre-processing task. Motion flow is extracted using Lucas Kanade optical flow algorithm (Sharmin, and Brad, 2012). The proposed *motion flow based Incept_LSTM* is able to detect the normal and abnormal behaviour of the person through joint learning of motion and appearance features. These features undergo a data fusion technique after being extracted from Inception v3 model. Such a combination of parameters and fusion is undertaken with transfer learning.

Transfer learning (Xing, Lv, Wang, Cao, Velenis, and Wang, 2019) is a technique in which knowledge obtained from one task can be applied to other. This task can be accomplished using pre-trained image networks. It can be performed in two different approaches. First, fine-tuning approach, in which pre-trained network is used and all the parameters of model are updated for performing new task. Second, the final layer is updated from where predictions are derived. This method of transfer-learning is known as feature extraction.

Pre-trained CNN is used as feature extractor and final layer is updated as per the required classification task. Once all the features have been obtained, the next step is feeding the LSTM network with those features. It extracts the temporal information from sequential data.

Feature Extraction using Pre-trained Model

DL models plays a significant role in extracting the features from image/video data. CNN (Xia, Huang and Wang, 2020) (Ahmad, Kazmi and Ali, 2019) has the capability of learning deep features from the static images. It has ability to learn spatial features present in the individual image frames. Training a DL model for the image representation requires a large volume of training data. Further, requires high computational resources to handle huge data. To resolve this limitation, the proposed model leveraged the concept of transfer learning. In the proposed model, the weights of pre-trained Inception v3 model is used to accomplish the task. Inception v3 is pre-trained model and has shown excellence in the area of image classification. Various other pre-trained model has shown remarkable results in this area. Due to its deep structure and top-5 accuracy score and reduced error rate, it has been chosen for the feature extraction process.

Motion Information

To understand the human activities from videos, it is required to analyse given data in spatial and temporal domain. As Video is a collection of static images operating in specific temporal range. Further, information in video data is not only processed spatially but sequence of frames is also considered for the correct understanding of the event/scene. This extra bit of motion information makes the problem more challenging.

There is a numerous increase in the number of parameters while considering video data. Therefore, it is a very challenging to design algorithm for detecting temporal structure in the given data with large number of parameters. This task can be accomplished by converting 2D networks to 3D networks to inherent the motion information. But this approach is not computationally effective. Videos are treated as the collection of separate frames and most methods learn features from the image frame only. And most of the classification is also performed frame wise. Maximum voting for one label is considered as the final classification for the video files, which is not always true. For e.g., a person throwing a ball in one frame may be misinterpreted by catching a ball in another frame. To understand the importance of the representation, it is required to estimate the motion in between consecutive frames.

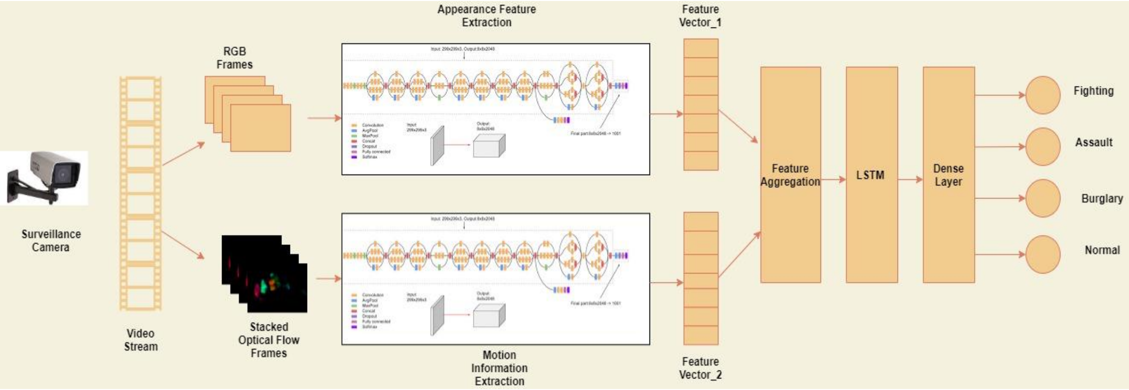


Figure 5.1: Schematic Representation of Motion flow based Incept_LSTM

To understand and estimate the motion information, optical flow is used. It is widely used in classifying videos at very low computational cost. Optical flow shows the direction of motion of the object in corresponding image frame. It works on the phenomenon of estimating the pixel brightness across the screen over time.

Optical flow estimation, in the proposed model, relies on following two assumptions (Savian, Elahi and Tillo, 2020):

1: Pixel intensity doesn’t change along the motion trajectory

2: Motion appears locally as a translation or neighbouring pixel have similar motions.

Variational method is one of the earliest methods of estimating the optical flow. It works on estimating the brightness over consecutive frames. It was the most simple and effective approach given by Horn Schunck (Savian, Elahi and Tillo, 2020). Another approach is given by Lucas Kanade (LK) (Mliki, Bouhlel and Hammami,2020), which works on considering consecutive frames and dividing them into patches of fixed sizes.

Feature Fusion

Motion and appearance features are fused together using feature level fusion to form a single feature vector. Aggregated feature vector is then passed further for processing. Independent feature vectors are combined together using late fusion (Xu, Yan, Ricci and Sebe, 2017) strategy to form a strong feature vector. The feature level fusion helps model to learn more effectively and jointly using two different or similar kind of modalities.

Sequential Learning Using LSTM

After extraction of these consecutive time-varying features, LSTM is used to accumulate the dynamic behaviour.

5.4 Experimental Results and Analysis

In proposed motion flow based Incept_LSTM, Experiments are conducted on segmented videos taken from UCF-Crime dataset. Data collected is pre-processed for further processing. Augmentation techniques are incorporated to enhance the size of the dataset. This method is really helpful when the size of the data is limited. Further, in order to train the model, experiments are carried out to select best hyper-parameters values. With the careful selection of hyper-parameters, model is trained for the designated task. Finally, testing is performed to validate the model.

5.4.1 Settings in Training and Testing Phase

- **Pre-processing**

Data is pre-processed before feeding into the model for training. Pre-processing is the essential step for training process. Not all the categories of UCF-crime dataset are selected rather, due to computational resource limitation this work focuses on only 4 categories. Optical frames are generated using Lucas Kanade algorithm for individual category. RGB and optical frames are used for the training purpose. These extracted frames were then combined in a group of 16 frames each to create one image of dimension 2048 X 128.

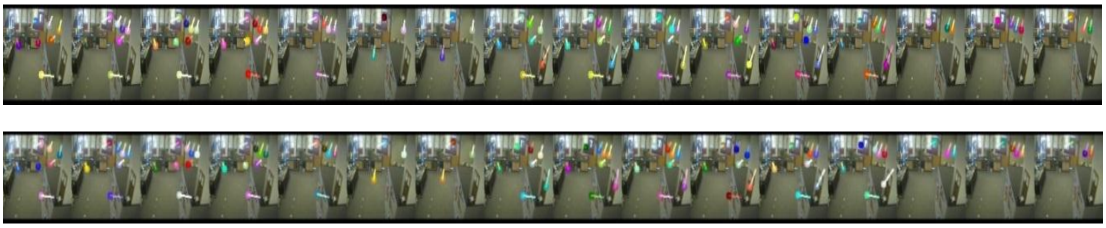


Figure 5.2: Optical Flow Computation Using Lucas Kanade Method

- **Hardware and Software Requirements**

For training and validation purpose following resources are selected:

Software Resources

TensorFlow 2.0: TensorFlow 2.0 (https://www.tensorflow.org) is an open-source, free library for developing DL models more efficiently. It is a tight integration of TensorFlow and Keras. It provides a high level API *tf.keras* to build neural networks and other ML models.

Python 3.8*:* Python (https://www.python.org/downloads/release/python-370/) offers a simple syntax for writing the program, which makes more comfortable reading and understanding of the code and also cost of maintenance is also reduced. It is an interactive language which has capability of interpreting and object-oriented programming concepts. High-level data structures are built-in functionality, supports dynamic linking which helps programmers to build rapid applications.

Other than these two major software tools, some more packages are used to fulfil the need of the problem. OpenCV, sklearn, numpy and pandas are used to accomplish the task.

Hardware Requirements

For this exploratory study, training and validation use available hardware resource with following specifications:

- Intel Core i7 9th Generation.
- Nvidia GTX
- Windows 10 Operating System
- 16 GB Ram

- **Model Training**

The model is trained in fully supervised manner. Each layer present in the architecture has some weight and bias which is initialized by randomly selected values. I trained this model for couple of times to observe the best hyper-parameters settings. Each time the values of parameters are changed and observed. For measuring the accuracy, categorical accuracy metric is used. Cross entropy is used as a loss function. It is used to measure the difference between real and probability distribution for a given random variable. In this work, cross entropy loss function is used to measure the error between a true and a predicted value. Purpose is to minimize the loss function, therefore, during the training process, the value of weight/parameter is also updated. During the training phase, batch size is set to 4 and epochs are set to 15. Furthermore, a small learning rate of 1e-6 is set to enhance the learning and fitting ability.

5.4.2 Analysis

The hyperparameter settings for proposed motion flow based Incept_LSTM are, the value of learning rate is 1e-6, activation function chosen is Leaky ReLU and the RMSProp is selected as optimizer.

5.5 Hyper-Parameters' impact on Model Performance

Hyper-parameters play a significant role on the performance of the classification model. Hyper-parameter such as type of optimizer used, learning rate and type of activation function show a significant impact on model performance. The experiment is implemented on UCF-Crime dataset and model performance is evaluated by changing the values of hyper-parameters.

Effect of optimizer: Goal of computation is to minimize the cost function by finding out the optimal value for weights. The function of optimizer is to calculate and update the parameters of the network. It affects the training and output from the model. Thus, it is suggested to carefully choose the optimizer for the training of deep networks. Adagrad, SGD, Adam and RMSprop are common optimizers. Performance of these optimizers are experimentally verified in table 5.1. Figure 5.3 shows the impact and performance of various selected optimizers for experiment. It can be seen from the figure5.4 that the model trained by RMSProp optimizer has a great fitting effect. Also, the gradient descent is not varying, rather giving the most stable curve. Hence, RMSProp is selected as the best optimizer for training the Incept_LSTM model.

Table 5.1: Performance Comparison on Various Optimizers

Optimizer	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss
Adagrad	48.8	1.202	21.66	1.449
RMS Prop	**98.2**	**0.2**	**94.57**	**0.3891**
Adam	91.8	0.4197	93.37	0.5630
SGD	92.9	0.4793	46.8	0.4835

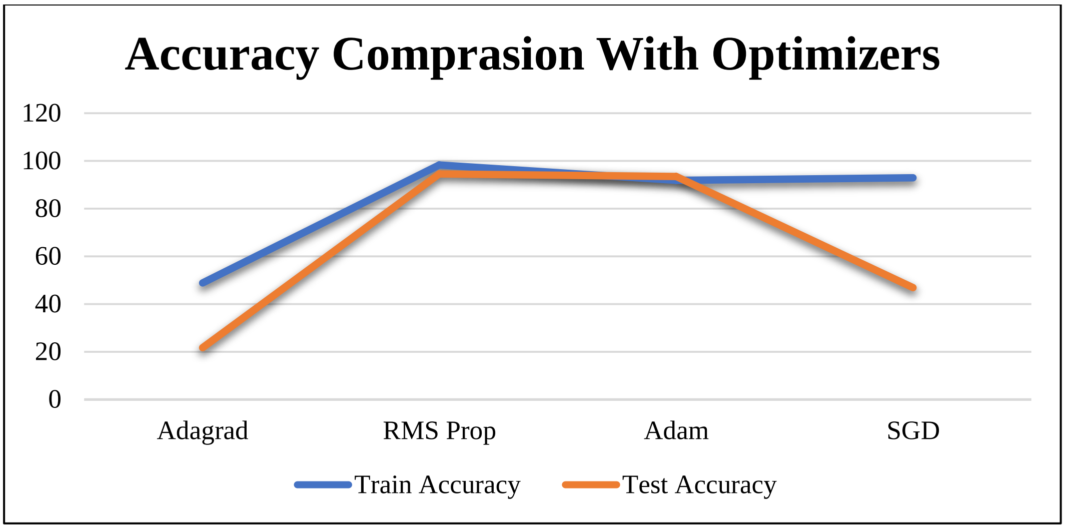


Figure 5.3: Accuracy Comparison with Various Optimizers

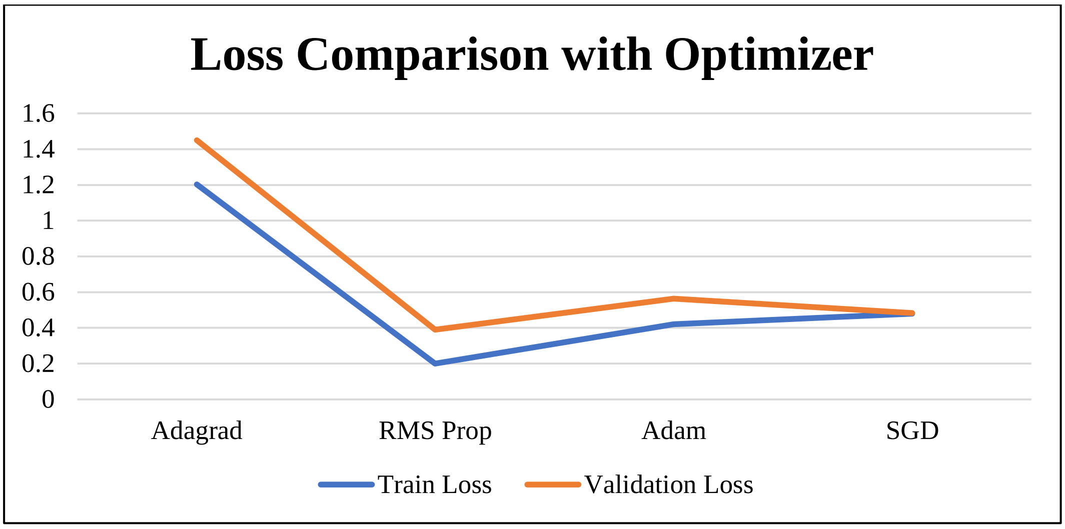


Figure 5.4: Loss Comparison with Various Optimizers

Effect of Activation Function: An activation function in neural network is responsible for transforming the weighted sum of the inputs to output node. These functions are attached to each neuron in the network and it determines whether it should be activated or not. The decision is made on analysing the neuron's current input, and sent further if it is helpful in predicting the results. Table 5.2 shows the comparison of using various activation functions. Figure 5.5 and figure 5.6 shows the accuracy and loss comparison with different activation functions.

Table 5.2: Performance Comparison on Various Activation Function

Activation Function	Accuracy	Loss
ReLu	97.6	0.3581
Leaky ReLu	**98.2**	**0.2**
Softmax	89.3	0.4547

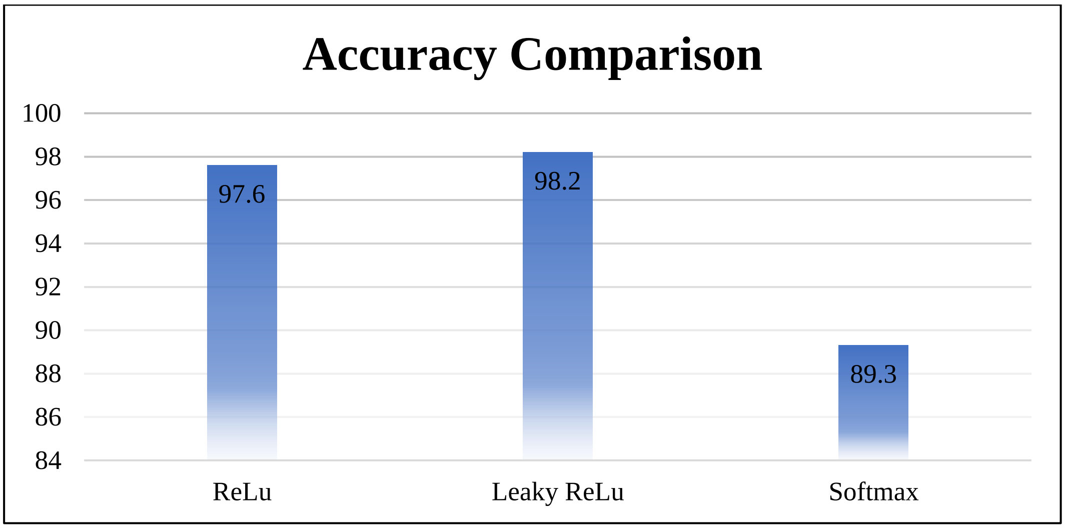


Figure 5.5: Accuracy Comparison on Various Activation Function

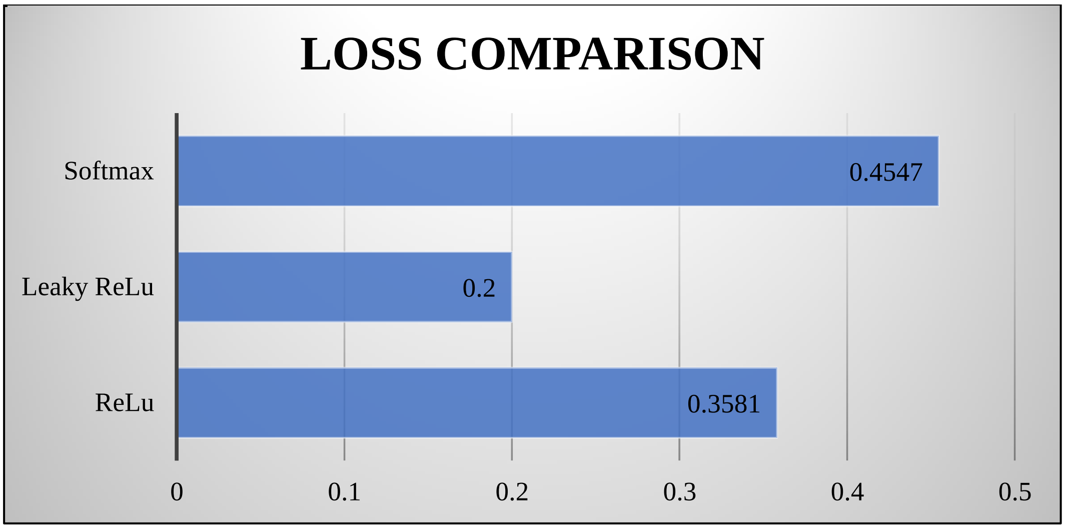


Figure 5.6: Loss Comparison on Various Activation Function

Effect of Learning Rate: Learning rate is defined as the step size for parameter update in training process. It is a hyper-parameter that controls the adjustment of the weights with respect to gradient loss. The value of learning rate lies in the range of 0.0 to 1.0. Smaller learning rates requires more training epochs for any update and it may happen that process may stuck in between. Whereas, the larger value of learning rate converges at faster speed and produces suboptimal results with a smaller number of epochs. It is very challenging for DL models for the careful selection of the learning rates.

Hyper-parameter tuning is performed to improve the training and testing loss. Instead of using fixed value for learning rate, it is suggested that with each iteration, the value of learning rate needs to be revised until an optimal solution is achieved.

For optimizing the performance of the proposed method, learning rate is tuned and analysed in table 5.3 On empirically testing, the model was able to learn the problem well with learning rates 1e-4 and 1e-6. Selection of too high (1e-2) and too low (1e-8) value of learning rate gives comparatively low model performance on train and test sets. On analysing, it is observed that 1e-6 gives the best possible outcome on train and test sets. Training and validation accuracy is achieved 98.2 and 94.57 respectively, also train and test loss incurred is minimum as compared to other empirically tested values. Figure 5.7 and 5.8 shows the accuracy and loss comparison on empirically set values of learning rates.

Table 5.3: Performance Comparison on Various Learning Rate

Learning Rate	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss
1e-2	85.2	0.5471	32.6	0.975
1e-4	92.15	0.4078	58.21	0.586
1e-6	**98.2**	**0.2**	**94.57**	**0.389**
1e-8	85.45	0.328	86.18	0.418

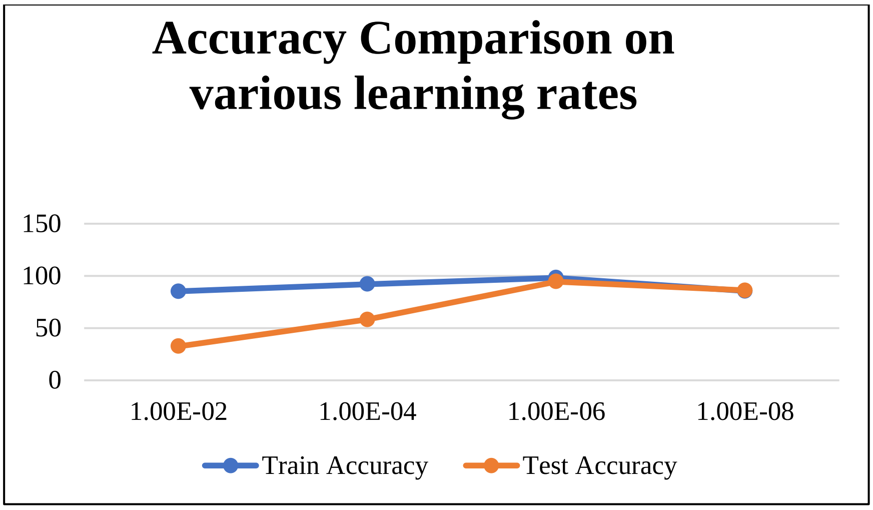


Figure 5.7: Accuracy Comparison on Various Learning Rates

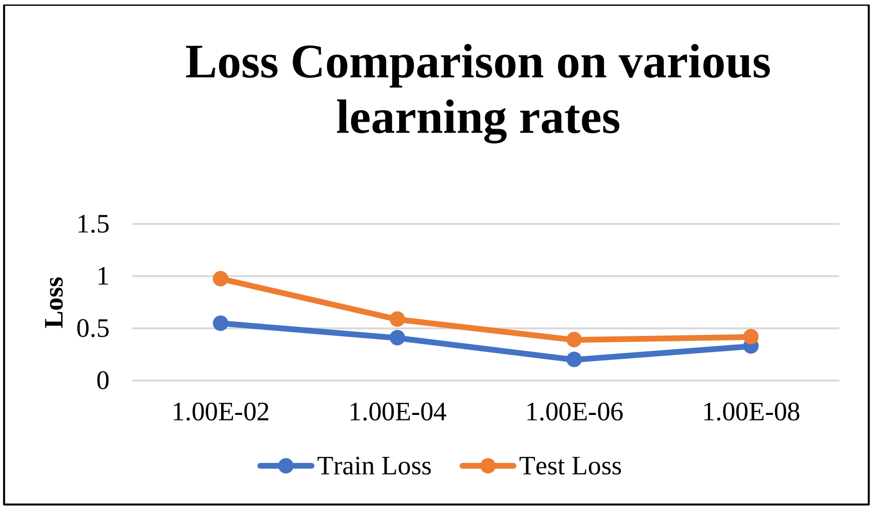


Figure 5.8: Loss Comparison on Various Learning Rates

Table 5.4: Motion flow based Incept_LSTM Results

Dataset	Epochs	Learning rate	Optimizer	Training accuracy (%)	Training Loss (%)	Validation Accuracy (%)	Validation Loss (%)	No. of output classes
UCF Crime dataset	**15**	**1e-6**	**RMSProp**	**98.2**	**0.2**	**94.57**	**0.38**	**4**

5.6 Comparison with state-of-the-art methods

In this section, the performance of the proposed model *motion flow based Incept_LSTM* is compared with state-of-the-art-methods mainly on UCF-Crime dataset. In recent times, DL models are performing very well, however, with its deep structure, the storage and computational requirement is also increasing.

In anomaly recognition, a delayed response can cause the loss of human being, and important assets. Therefore, model selection for *feature extraction* is important for anomaly detection. Table 5.5 presents the accuracy comparison of the proposed model with the state-of-the-art methods.

Table 5.5: Accuracy comparison with State-of-the-art methods

Year	Method used	Accuracy	Dataset Used	Reference
2018	DEARESt based on VGG19 + FlowNet for motion flow	76.66%	UCF-Crime dataset	(Biradar, Dube and Vipparthi, 2018)
2018	VGGNet+Bi-directional LSTM	81%	UCF-Crime dataset	(Nair, L.K., 2018)
2021	VGG16 + BD-LSTM	82/87.5%	UCF-Crime dataset/UCFCrime2Local	(Ullah, Ullah,Haq, Muhammad, Sajjad and Baik, 2021)
2021	Inception v3+BD-LSTM	80/88%	UCF-Crime dataset/UCFCrime2Local	(Ullah, Ullah,Haq, Muhammad, Sajjad and Baik, 2021)
2021	Resnet50+BD-LSTM	85.53/89.05%	UCF-Crime dataset/UCFCrime2Local	(Ullah, Ullah,Haq, Muhammad, Sajjad and Baik, 2021)
2021	Inception v3+LSTM	88.37%	KTH	(Begampure and Jadhav, 2021)
2021	CNN(MobileNet)+Attention based LSTM	78.30%	UCF-Crime Dataset	(Ulla, Ullah, Hussain, Khan and Baik, 2021)
2022	**Proposed Method**	**94.57%**	UCF-Crime Dataset	-

In the proposed model, *Motion flow based Incept_LSTM*, the number of parameters is 23 million, with a 1.5 GB model size. The time taken to process a sequence of 16 frames is 0.08 seconds with a total training time of 17hrs 13minutes.

Table 5.6 presents the comparison on the number of parameters and processing time taken by the individual model to process a sequence of frames for activity recognition.

Table 5.6: Comparison of the proposed method with state-of-the-art in terms of the parameters and time complexity

Year	Method Used	No. of Parameters (in Millions)	Time Complexity/Per Sequence (in seconds)	References
2018	DEARESt based on VGG19 + FlowNet for motion flow	305.49	-	(Biradar, Dube and Vipparthi, 2018)
2021	VGG16 + BD-LSTM	143	0.22	(Ullah, Ullah,Haq, Muhammad, Sajjad and Baik, 2021)
2021	Inception v3 + BD-LSTM	23	-	(Ullah, Ullah,Haq, Muhammad, Sajjad and Baik, 2021)
2021	Resnet50+BD-LSTM	25	0.2	(Ullah, Ullah,Haq, Muhammad, Sajjad and Baik, 2021)
2022	**Proposed Approach**	**23**	**0.08**	**-**

5.7 Summary

In this chapter, the baseline Incept_LSTM is optimized. To improve the recognition accuracy, the importance of motion flow with video data is analysed and incorporated. The input to baseline model has changed and provided with the more construct, required for the video data. Further, impact of hyper-parameters is studied and analysed. An empirical driven approach is used to optimize Incept_LSTM by adjusting the values of these parameters. The impact of learning rate, activation function and optimizers are carefully observed. And empirically observed results are used for the final training purpose. To establish the validity of the results, the final computed results are compared with state-of-the-art accuracies.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 Conclusion

With the increase in crime rate, there is a massive increase in the installation of surveillance cameras. In order to ensure public safety, various CCTVs have been installed in the areas like airport, railway station, colleges, schools, day-cares, malls, parks and other public places. Monitoring these surveillance cameras on 24x7 basis, requires a lot of manpower. The accuracy of the system depends upon the activeness of the supervising entity. Sometimes, biased nature of the supervising person, fatigue issue results in the inaccurate decision. Therefore, there is a dire need of automated surveillance system, that can detect the anomalous event correctly and in time. The problem of detecting anomalous event is actually recognizing the human activity closely from the data collected through CCTVs. Analysing a single image frame doesn’t provide the accurate results. Monitoring single image frame may give different pattern in subsequent frame. So, to understand the need of the system, emphasis is given on the sequence of the data and its temporal connection.

Feature Extraction is foremost and important step for any recognition model. Traditional ML algorithms requires human efforts for the feature extraction. For specific problem, domain knowledge and human experience is the key for the success of any classification and recognition model. Ability of DL models makes it more acceptable in various domains. DL models have capability to automatically adjust the parameters as per the training dataset. Visual features from the training and test data are extracted in end-to-end way. In this work, DL is brought to train the model for recognizing human activities. However, it has certain limitation like

- deep models are generally designed by trial-and-error process, which requires large amount of labelled training data
- Also, large number of neuron connections will result into large computational expense.

Therefore, Transfer Learning technique is used to handle the mentioned limitations. This task can be accomplished using pre-trained image networks. It can be performed via two different approaches. First, fine-tuning approach, in which pre-trained network is used and all the parameters of model are updated for performing new task. Second, the final layer is updated from where predictions are derived. This method of transfer-learning is known as feature extraction.

CNN is known to best for image recognition, classification and pattern analysis. CNN structure and its various architectures have been studied and observed in *chapter-3*. Lenet is one of the simplest networks with 2 convolutional and 3 fully connected layer. It has approx.60,000 parameters. Another architecture introduced in 2012, AlexNet, with 8 layers (5 convolutional and 3 fully connected) with 60 million parameters. It was the first one, who implemented ReLU activation function. Further, VGG-16 came into existence in 2014. It consists of approximately 138 million parameters. The design of VGG is deeper as compared to AlexNet. Furthermore, Inception and its variants came into existence. Due to its deeper structure and more varying size filters makes it more acceptable for extracting complex features. Inception-v3 has one auxiliary classifier attached in between. Next, design of ResNet supports deeper structure without compromising model's generalization power. This analysis gives a strong baseline for selecting the pre-trained network for feature extraction.

Further, Chapter-4 focuses on the methodology designed for HAR. It is inaccurate to give decision for ongoing event by observing only single image frame. It may happen that event occurring in one frame is different in another subsequent frames. So, it is important to consider the temporal domain of the data. In chapter-4, Incept_LSTM is designed for monitoring the human activities. Inception v3 is chosen for the hidden feature extraction and LSTM is used for maintaining the temporal relationship between the frames. Stacking of two LSTM layer is used to improvise the accuracy of the model. The model proposed in said section is trained with UCF-Crime dataset. Though, the dataset is very huge and contains 14 classes with 1 normal and 13 abnormal classes of activities. Due to computational limitations, experiments are performed on only 4 classes *(normal, assault, burglary and fighting)* of activities. The accuracy achieved by Incept_LSTM is 91%. The result of the Incept_LSTM (*accuracy and F1-score*) is compared with state-of-the-art.

There is an increase in the number of parameters while considering video data. Therefore, it is a very challenging to design algorithm for detecting temporal structure in the given data with large number of parameters. Sometimes, video contains redundant information which makes the training process little slower and capability to extract unique features is also get reduced. Besides, with only RGB frames, it is difficult to capture the temporal information from the video data. A vast work is done in the literature to process videos as RGB sequence with 2D CNN, 3D CNN and RNN or by converting 2D networks to 3D networks to inherent the motion information. But this approach is not computationally effective. Videos are treated as the collection of separate frames and most methods learn features from the image frame only. To consider temporal data more effectively, motion flow is computed and incorporated in *Chapter-5*. An motion flow based Incept_LSTM is designed with two input streams. The extracted features are fused together to make a single rich feature vector which further passes to LSTM for handling the dynamic behaviour of the frames. Finally, motion flow based Incept_LSTM is able to perform multiclass classification. The model is trained and tested on UCF-Crime dataset. Further, role and importance of hyperparameters are discussed in chapter-5. Various change hyper-parameters settings (*optimizer, learning rate, activation function*) are performed and observed. Finally, the model is trained with those empirically settings which results in good accuracy score and minimum loss. Further, to validate the results, a comparison is made with the existing work present in the literature. A comparison is mainly performed with those, who opted UCF-Crime dataset and one comparison with KTH dataset also. On comparison, motion flow based Incept_LSTM approach outperforms the methodologies present in the literature. Further, the dynamics of proposed model is computed: model parameters, model size and time. Although, the time complexity parameter is totally hardware dependent. One having GPU of higher configuration may have completed the task earlier.

6.2 Future Scope

Video based surveillance has gained much attention due to its wide applicability. It has also been established in literature that an automatic surveillance decreases the crime rate to a significant level. In the present work, focus has been on understanding and analysing videos for detecting abnormal behaviour (broadly defined in four classes) using DL models. Although the proposed work has achieved significant accuracy as compared to state-of-the-art literature, there is still scope of improvement. In the current context, the proposed work analysed the behaviour of a person in terms of activities performed during a certain time span. However, automatic surveillance has many dimensions which are still unexplored or less explored. The factors considered for detecting abnormal behaviour can be further exploited with factors like, demographics, vicinity where a crime is conducted, intentions, and personal enmity. Face recognition and detection of observable nodal points can also help in understanding the behaviour. This may lead to better understanding of the situation and thus can be investigated more deeply with other soft computing techniques like genetic algorithm, fuzzy logic, type-1 fuzzy logic, type-2 fuzzy logic. In order to understand more complex emotions of the humans, and make automatic surveillance efficient it is required to identify and analyse specific scenes from pixel analysis also called as automatic scene understanding. Further work can be done on finding a reliable description and classification of such scenes. The proposed work can be extended for better accuracy, real-time execution, computational complexity and better adaptation of application.

REFERENCES

[1] Abiodun, O.I., Jantan, A., Omolara, A.E., Dada, K.V., Mohamed, N.A. and Arshad, H., 2018. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, *4*(11), p.e00938.

[2] Albawi, S., Mohammed, T.A. and Al-Zawi, S., 2017, August. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). Ieee.

[3] Aghaei, A., Nazari, A. and Moghaddam, M.E., 2021. Sparse Deep LSTMs with Convolutional Attention for Human Action Recognition. *SN Computer Science*, *2*(3), pp.1-14.

[4] Arifoglu, D. and Bouchachia, A., 2017. Activity recognition and abnormal behaviour detection with recurrent neural networks. *Procedia Computer Science*, *110*, pp.86-93.

[5] Ahmad, W., Kazmi, B.M. and Ali, H., 2019, December. Human activity recognition using multi-head CNN followed by LSTM. In *2019 15th international conference on emerging technologies (ICET)* (pp. 1-6). IEEE.

[6] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., 2021. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*(1), pp.1-74.

[7] Biradar, K., Dube, S. and Vipparthi, S.K., 2018, December. DEARESt: Deep Convolutional aberrant behavior detection in real-world scenarios. In *2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS)* (pp. 163-167). IEEE.

[8] Begampure, S. and Jadhav, P., 2021. Intelligent video analytics for human action detection: a deep learning approach with transfer learning. *International Journal of Computing and Digital System*.

[9] Chowdhury, A.K., Tjondronegoro, D., Chandran, V. and Trost, S.G., 2017. Physical activity recognition using posterior-adapted class-based fusion of multiaccelerometer data. *IEEE journal of biomedical and health informatics*, *22*(3), pp.678-685.

[10] Dang, L.M., Min, K., Wang, H., Piran, M.J., Lee, C.H. and Moon, H., 2020. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, *108*, p.107561.

[11] Dubey, S., Boragule, A., Gwak, J. and Jeon, M., 2021. Anomalous Event Recognition in Videos Based on Joint Learning of Motion and Appearance with Multiple Ranking Measures. *Applied Sciences*, *11*(3), p.1344.

[12] Fortune Business Insights, Video Surveillance Market Size, Share, Growth | Industry Trends, 2026 “fortunebusinessinsights.com”, last accessed on: 17.12.21

[13] Gao, J., Yang, Y., Lin, P. and Park, D.S., 2018. Computer vision in healthcare applications.

[14] Goodfellow, I., Bengio, Y. and Courville, A., 2016. *Deep learning*. MIT press

[15] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T., 2018. Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, pp.354-377.

[16] Georgiou, T., Liu, Y., Chen, W. and Lew, M., 2020. A survey of traditional and deep learning-based feature descriptors for high dimensional data in computer vision. *International Journal of Multimedia Information Retrieval*, *9*(3), pp.135-170.

[17] Girdhar, P., Johri, P. and Virmani, D., 2020. Incept_LSTM: Accession for human activity concession in automatic surveillance. *Journal of Discrete Mathematical Sciences and Cryptography*, pp.1-15.

[18] Ha, S. and Choi, S., 2016, July. Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors. In *2016 International Joint Conference on Neural Networks (IJCNN)* (pp. 381-388). IEEE.

[19] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[20] Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.

[21] Hu, G., Yang, Y., Yi, D., Kittler, J., Christmas, W., Li, S.Z. and Hospedales, T., 2015. When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. In *Proceedings of the IEEE international conference on computer vision workshops* (pp. 142-150).

[22] Ha, S., Yun, J.M. and Choi, S., 2015, October. Multi-modal convolutional neural networks for activity recognition. In *2015 IEEE International conference on systems, man, and cybernetics* (pp. 3017-3022). IEEE.

[23] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[23] He, X., Zhao, K. and Chu, X., 2021. AutoML: A Survey of the State-of-the-Art. *Knowledge-Based Systems*, *212*, p.106622.

[24] Imran, H.A. and Latif, U., 2020, December. HHARNet: Taking inspiration from Inception and Dense Networks for Human Activity Recognition using Inertial Sensors. In *2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)* (pp. 24-27). IEEE.

[25] Ilse, M., Tomczak, J.M. and Forré, P., 2021, July. Selecting data augmentation for simulating interventions. In *International Conference on Machine Learning* (pp. 4555-4562). PMLR.

[26] Jahne B., Haubecker H.,2000, *Computer Vision and Applications: A Guide for students and practioners,* Academic Press, San Dieago,pp. 1-8(512)

[27] Jain, A.K. and Li, S.Z., 2011. *Handbook of face recognition* (Vol. 1). New York: springer.

[28] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, pp.1097-1105.

[29] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, pp.1097-1105.

[30] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), pp.2278-2324.

[31] Luo, X., Li, H., Cao, D., Yu, Y., Yang, X. and Huang, T., 2018. Towards efficient and objective work sampling: Recognizing workers' activities in site surveillance videos with two-stream convolutional networks. Automation in Construction, 94, pp.360-370.

[32] Murad, A. and Pyun, J.Y., 2017. Deep recurrent neural networks for human activity recognition. *Sensors*, *17*(11), p.2556.

[33] Ma, C.Y., Chen, M.H., Kira, Z. and AlRegib, G., 2019. TS-LSTM and temporal-inception: Exploiting spatiotemporal dynamics for activity recognition. *Signal Processing: Image Communication*, *71*, pp.76-87.

[34] Mustafa, T., Dhavale, S. and Kuber, M.M., 2020. Performance Analysis of Inception-v2 and Yolov3-Based Human Activity Recognition in Videos. *SN Computer Science*, *1*(3), pp.1-7.

[35] Minnen, D., Westeyn, T., Starner, T., Ward, J.A. and Lukowicz, P., 2006. Performance metrics and evaluation issues for continuous activity recognition. *Performance Metrics for Intelligent Systems*, *4*, pp.141-148.

[36] Mliki, H., Bouhlel, F. and Hammami, M., 2020. Human activity recognition from UAV-captured video sequences. Pattern Recognition, 100, p.107140.

[37] Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S., 2018. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.

[38] Noori, F.M., Wallace, B., Uddin, M.Z. and Torresen, J., 2019, June. A robust human activity recognition approach using openpose, motion features, and deep recurrent neural network. In *Scandinavian conference on image analysis* (pp. 299-310). Springer, Cham.

[39] Nair, L.K., 2018. PREDICTION OF ANOMALOUS ACTIVITIES IN A VIDEO.

[40] Nasaruddin, N., Muchtar, K., Afdhal, A. and Dwiyantoro, A.P.J., 2020. Deep anomaly detection through visual attention in surveillance videos. *Journal of Big Data*, *7*(1), pp.1-17.

[41] Ordóñez, F.J. and Roggen, D., 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors, 16(1), p.115.

[42] Pandey, N.N. and Muppalaneni, N.B., 2021. Temporal and spatial feature-based approaches in drowsiness detection using deep learning technique. Journal of Real-Time Image Processing, pp.1-13.

[43] Peng, S., Huang, H., Chen, W., Zhang, L. and Fang, W., 2020. More trainable inception-ResNet for face recognition. *Neurocomputing*, *411*, pp.9-19.

[44] Pareek, P. and Thakkar, A., 2021. A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artificial Intelligence Review*, *54*(3), pp.2259-2322

[45] Python_https://www.python.org/downloads/release/python-370/

[46] Rosenfeld, A., 1997. Image analysis and computer vision: 1996. *Computer Vision and Image Understanding*, *66*(1), pp.33-93.

[47] Ronao, C.A. and Cho, S.B., 2016. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert systems with applications*, *59*, pp.235-244.

[48] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *201*.

[49] Rao, B.V.K., Gopikrishna, K.S., Sukrita, M. and Parameswaran, L., 2021. Activity Recognition Using LSTM and Inception Network. In *Soft Computing and Signal Processing* (pp. 119-128). Springer, Singapore.

[50] Shapiro, L.G., 2020. Computer vision: the last 50 years. *International Journal of Parallel, Emergent and Distributed Systems*, *35*(2), pp.112-117.

[51] Sultani, W., Chen, C. and Shah, M., 2018. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6479-6488).

[52] Sargano, A.B., Angelov, P. and Habib, Z., 2017. A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition. *applied sciences*, *7*(1), p.110.

[53] Singh, D., Merdivan, E., Psychoula, I., Kropf, J., Hanke, S., Geist, M. and Holzinger, A., 2017, August. Human activity recognition using recurrent

neural networks. In *International cross-domain conference for machine learning and knowledge extraction* (pp. 267-274). Springer, Cham.

[54] Sibi, P., Jones, S.A. and Siddarth, P., 2013. Analysis of different activation functions using back propagation neural networks. *Journal of theoretical and applied information technology*, *47*(3), pp.1264-1268.

[55] Sharma, S., Sharma, S. and Athaiya, A., 2017. Activation functions in neural networks. *towards data science*, *6*(12), pp.310-316.

[56] Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural networks*, *61*, pp.85-117.

[57] Suresh, A.J. and Visumathi, J., 2020. Inception ResNet deep transfer learning model for human action recognition using LSTM. *Materials Today: Proceedings*.

[58] Sarma, N., Chakraborty, S. and Banerjee, D.S., 2019, January. Activity recognition through feature learning and annotations using LSTM. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)* (pp. 444-447). IEEE.

[59] Sarma, N., Chakraborty, S. and Banerjee, D.S., 2019, January. Learning and Annotating Activities for Home Automation using LSTM. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)* (pp. 631-636). IEEE.

[60] Shorten, C. and Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), pp.1-48.

[61] Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[62] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions.

In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[63] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. and Wojna, Z., 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818-2826).

[64] Staudemeyer, R.C. and Morris, E.R., 2019. Understanding LSTM--a tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv preprint arXiv:1909.09586*.

[65] Sharmin, N. and Brad, R., 2012. Optimal filter estimation for Lucas-Kanade optical flow. *Sensors*, *12*(9), pp.12694-12709.

[66] Savian, S., Elahi, M. and Tillo, T., 2020. Optical flow estimation with deep learning, a survey on recent advances. In Deep biometrics (pp. 257-287). Springer, Cham.

[67] Turaga, P., Chellappa, R., Subrahmanian, V.S. and Udrea, O., 2008. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video technology*, *18*(11), pp.1473-1488.

[68] Tensor-Flow_*https://www.tensorflow.org*

[69] Umbaugh, S.E., 2010. *Digital image processing and analysis: human and computer vision applications with CVIPtools*. CRC press.

[70] Ucf-crime dataset (real-world anomalies detection in videos), https://webpages.uncc.edu/cchen62/dataset.html, June 2019.

[71] Ullah, W., Ullah, A., Haq, I.U., Muhammad, K., Sajjad, M. and Baik, S.W., 2021. CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks. *Multimedia Tools and Applications*, *80*(11), pp.16979-16995.

[72] Ullah, W., Ullah, A., Hussain, T., Khan, Z.A. and Baik, S.W., 2021. An Efficient Anomaly Recognition Framework Using an Attention Residual LSTM in Surveillance Videos. *Sensors*, *21*(8), p.2811.

[73] Vrigkas, M., Nikou, C. and Kakadiaris, I.A., 2015. A review of human activity recognition methods. *Frontiers in Robotics and AI*, *2*, p.28.

[74] Venkata B., Krishna RaoK., GopikrishnaM. S andSukritaLatha Parameswaran, 2021,"Activity Recognition Using LSTM and Inception Network",Soft Computing and Signal Processing, pp 119-128.

[75] Weng, Z., Li, W. and Jin, Z., 2021. Human activity prediction using saliency-aware motion enhancement and weighted LSTM network. *EURASIP Journal on Image and Video Processing*, *2021*(1), pp.1-23.

[76] Wang, D., Candinegara, E., Hou, J., Tan, A.H. and Miao, C., 2017, December. Robust human activity recognition using lesser number of wearable sensors. In *2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)* (pp. 290-295). IEEE.

[77] Weiss, K., Khoshgoftaar, T.M. and Wang, D., 2016. A survey of transfer learning. *Journal of Big data*, *3*(1), pp.1-40.

[78] Wang, S., Sun, S. and Xu, J., 2015. Auc-maximized deep convolutional neural fields for sequence labeling. *arXiv preprint arXiv:1511.05265*.

[79] Xu, C., Chai, D., He, J., Zhang, X. and Duan, S., 2019. InnoHAR: A deep neural network for complex human activity recognition. *Ieee Access*, *7*, pp.9893-9902.

[80] Xia, K., Huang, J. and Wang, H., 2020. LSTM-CNN architecture for human activity recognition. *IEEE Access*, *8*, pp.56855-56866.

[81] Xu, Q., Zhang, M., Gu, Z. and Pan, G., 2019. Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs. *Neurocomputing*, *328*, pp.69-74.

[82] Xu, D., Yan, Y., Ricci, E. and Sebe, N., 2017. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, *156*, pp.117-127.

[83] Xing, Y., Lv, C., Wang, H., Cao, D., Velenis, E. and Wang, F.Y., 2019. Driver activity recognition for intelligent vehicles: A deep learning approach. *IEEE transactions on Vehicular Technology*, *68*(6), pp.5379-5390.

[84] Ye, H., Wu, Z., Zhao, R.W., Wang, X., Jiang, Y.G. and Xue, X., 2015, June. Evaluating two-stream CNN for video classification. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval* (pp. 435-442).

[85] Yu, T. and Zhu, H., 2020. Hyper-parameter optimization: A review of algorithms and applications. *arXiv preprint arXiv:2003.05689*.

[86] Zeng, M., Gao, H., Yu, T., Mengshoel, O.J., Langseth, H., Lane, I. and Liu, X., 2018, October. Understanding and improving recurrent networks for human activity recognition by continuous attention. In *Proceedings of the 2018 ACM international symposium on wearable computers* (pp. 56-63).

[87] Zebin, T., Scully, P.J. and Ozanyan, K.B., 2016, October. Human activity recognition with inertial sensors using a deep learning approach. In *2016 IEEE SENSORS* (pp. 1-3). IEEE.

[88] Zeiler, M.D. and Fergus, R., 2014, September. Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

[89] Zhu, F., Shao, L., Xie, J. and Fang, Y., 2016. From handcrafted to learned representations for human action recognition: A survey. *Image and Vision Computing*, *55*, pp.42-52.