

Feature Regularization and Learning for Human Activity Recognition



By

Osayamwen Festus Osazuwa

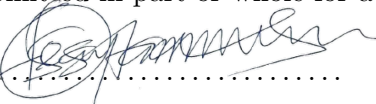
212562369

A thesis submitted in fulfillment of the academic requirements for the Degree of
Doctor of Philosophy in Computer Engineering in the School of Engineering
University of KwaZulu-Natal Durban, South Africa

2018 December

**UNIVERSITY OF KWAZULU-NATAL, COLLEGE OF
AGRICULTURE, ENGINEERING AND SCIENCE
DECLARATION**

The research described in this thesis was performed at the University of KwaZulu-Natal under the supervision of Prof. Jules-Raymond Tapamo. I hereby declare that all materials incorporated in this thesis are my own original work except where acknowledgement is made by name or in form of reference. The work contained herein has not been submitted in part or whole for a degree at any other university.

Signed: 
Festus Osazuwa Osayamwen
Date: December 2018

As the candidate's supervisor, I have approved/disapproved this dissertation for submission.


Signed:.....
Prof. Jules-Raymond Tapamo
Date: December 2018

**UNIVERSITY OF KWAZULU-NATAL, COLLEGE OF
AGRICULTURE, ENGINEERING AND SCIENCE
DECLARATION 1 -PLAGIARISM**

I, FESTUS OSAZUWA OSAYAMWEN, declare that:

1. The research reported in this thesis, except where otherwise indicated, is my original research.
2. This thesis has not been submitted for any degree or examination at any other university.
3. This thesis does not contain other persons data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.
4. This thesis does not contain other persons writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then
 - a. Their words have been re-written but the general information attributed to them has been referenced;
 - b. Where their exact words have been used, then their writing has been placed inside quotation marks, and referenced.
5. Where I have reproduced a publication of which I am an author, co-author or editor, I have indicated in detail which part of the publication was actually written by myself alone and have fully referenced such publications.
6. This thesis does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed.



**UNIVERSITY OF KWAZULU-NATAL, COLLEGE OF
AGRICULTURE, ENGINEERING AND SCIENCE
DECLARATION 2 -PUBLICATIONS**

I, FESTUS OSAZUWA OSAYAMWEN, declare that the following publications came out of this thesis

1. **F. Osayamwen** and J.R. Tapamo, Deep Learning Class Discrimination Based on Prior Probability for Human Activity Recognition, IEEE Access, Vol. 7, pp. 14747-14756, 2019.
2. **F. Osayamwen** and J.R. Tapamo, Improved Eigenspectrum Regularization for Human Activity Recognition, Int. J. Computational Vision and Robotics, Vol. 8, No. 4, 2018, pp. 435-454, 2018.
3. **F. Osayamwen** and J.R. Tapamo, Within-Class Subspace Regularization for Human Activity Recognition, in Proceedings of the 2016 IEEE International Conference on Industrial Technology (ICIT), Taipei, Taiwan, March 14-17, 2016, pp. 804-807, March 2016.

Signed.



Dedication

I dedicate this work to God almighty for his mercies and love towards me, to my wife, mum, brother, sisters and my two lovely children Prosper and Goodness. Also in remembrance of my father, late Mr. Wilfred Osayamwen, who thought me disciplines and life values.

Acknowledgements

I would like to acknowledge my supervisor and discipline leader, Department of Electrical, Electronic and Computer Engineering, Prof. Jules-Raymond Tapamo for his guidance, directions and financial support during the entire period of my doctoral research. I am highly indebted to the authority of the University of Benin and the federal government of Nigeria for their generosity in granting me a luxurious study leave and financial muscle to enable me to undertake this research. I also want to express my sincere gratitude to my lovely father late Mr. Wilfred Ajayi Osayamwen and mother Mrs Helen Osayamwen for their immense strength and encouragement that I received from them. To my late father, you were the source of my inspiration and your encouragement followed me through even in your glorious call. Many thanks to my amiable wife Praise Osayamwen and my two lovely children Prosper and Goodness. I thank you all for your understanding during my long absence from home when you needed me most. Special thanks to my brother Mr. Austin Osayamwen, and my two lovely sisters Mrs kuyenu Osamede, and Mrs Joy Omoruyi for their unfailing love and support in the journey so far. I like to acknowledge and appreciate my endless list of friends amongst who are, Dr. Andrew Eloka-Eboka, Dr. Uyi Igbinosa, Ireyuwa Igbinosa, Dr. Usiholo Irunasi, Mrs Kenalemang Nkwoji, Mr. Illama Oshiokayame, Mr. Soga Akinro and others who gave me moral and spiritual support during this research. Lastly, my sincere gratitude to almighty God who kept me alive, my source of light and strength whenever I am weak and tired. I say thanks be to God Almighty.

Abstract

Feature extraction is an essential component in the design of human activity recognition model. However, relying on extracted features alone for learning often makes the model a suboptimal model. Therefore, this research work seeks to address such potential problem by investigating feature regularization. Feature regularization is used for encapsulating discriminative patterns that are needed for better and efficient model learning. Firstly, a within-class subspace regularization approach is proposed for eigenfeatures extraction and regularization in human activity recognition. In this approach, the within-class subspace is modelled using more eigenvalues from the reliable subspace to obtain a four-parameter modelling scheme. This model enables a better and true estimation of the eigenvalues that are distorted by the small sample size effect. This regularization is done in one piece, thereby avoiding undue complexity of modelling eigenspectrum differently. The whole eigenspace is used for performance evaluation because feature extraction and dimensionality reduction are done at a later stage of the evaluation process. Results show that the proposed approach has better discriminative capacity than several other subspace approaches for human activity recognition. Secondly, with the use of likelihood prior probability, a new regularization scheme that improves the loss function of deep convolutional neural network is proposed. The results obtained from this work demonstrate that a well regularized feature yields better class discrimination in human activity recognition. The major contribution of the thesis is the development of feature extraction strategies for determining discriminative patterns needed for efficient model learning.

Contents

List of Figures	vii
List of Tables	ix
1 Introduction	1
1.1 Background	1
1.2 Research Motivation	6
1.2.1 The Main goal of the study	9
1.2.2 Specific Objectives	9
1.3 Thesis Overview	9
1.4 Contributions	10
2 Literature Review	12
2.1 A Brief Historical Review on Deep Learning	19
2.2 Building Deep Representational Learning	21
2.2.1 Related Works in Group Activity Recognition Using Deep Neural Network	25
2.3 Different Classes of Deep Learning architecture	27
2.3.1 Generative architecture	27
2.3.2 Discriminative architecture	29
2.4 Deep model Training and its challenges	31
2.5 Summary and Discussion	34
3 Image Detection, Extraction and Pre-processing	36
3.1 Introduction	36
3.2 Image detection method	37

3.2.1	Background Subtraction	39
3.3	The Image Enhancement Process	42
3.3.1	Extracting Silhouettes Images	45
3.3.2	Extracting Grayscale Images	45
3.4	Silhouettes Versus Grayscale Images	47
3.5	Feature Extraction and Dimensionality Reduction	48
3.5.1	Principal Component Analysis	49
3.5.2	Fisher Linear Discriminant Analysis	51
3.5.3	Silhouettes and grayscale feature evaluation on the weizmann dataset	53
3.6	Results obtained using PCA and FLDA on both form of motion infor- mation representation	53
3.7	Summary and Discussion	56
4	Improved Eigenspectrum Regularization for Human Activity Recog- nition	57
4.1	Introduction	57
4.2	Eigenspectrum Modeling	60
4.2.1	The Problem of unregularised subspace	62
4.2.2	Effect of Using Small Eigenvalues to Scaled Eigenvectors	63
4.2.3	Extrapolation and Modeling of Within-Class Matrix Using Four- Parameters	64
4.3	Regularization and Extraction of features	67
4.4	Experimental Result and Discussion	70
4.4.1	Results on the KTH Database	73
4.5	Summary and Discussion	76
5	Deep Learning and Feature regularization in Convolutional Architec- ture for Human Activity Recognition	77
5.1	Introduction	77
5.2	Architectural Design of Convolutional Neural Network	79
5.2.1	Convolution Layer	80
5.2.2	Locally Connected Network	81
5.2.3	Spatial arrangement of output volume	81

5.2.4	Rectified Unit	82
5.2.5	Sigmoid Activation Function	84
5.2.6	Tanh Activation Function	86
5.2.7	Pooling Layer	86
5.2.8	The Fully Connected Layer	87
5.3	Deep Learning Class Discrimination in Human Activity Recognition . .	87
5.3.1	Challenges of Deep learning Class Discrimination with CNN . . .	88
5.4	Loss Function and Convolutional Neural Networks Optimization	89
5.4.1	Joint Supervisory Loss	94
5.4.2	Our Proposed Method	97
5.5	Experimental and CNN Detailed Setup	100
5.5.1	Experimental results on Weizmann and KTH dataset	101
5.6	Model Optimization	106
5.6.1	Neural Network Dropout Model Description	108
5.6.2	Empirical Results on the Dropout Regularization	109
5.6.3	L2 feature Regulaization	120
5.6.4	Effect of L2 regularization on KTH dataset	122
5.7	Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization	123
5.7.1	Learning Process	126
5.8	Summary and Discussion	132
6	Conclusion and Future works	135
6.1	Summary of work	135
6.2	Future works	139
	References	140

List of Figures

1.1	Passive Cameras	7
1.2	Intelligent Camera	8
2.1	Different Human Action	22
2.2	Feature Extraction and Training	23
3.1	Background Subtraction for Video Information Recovery	40
3.2	Video Information Extraction	41
3.3	Image Preprocessing	43
3.4	Graphical Illustration of Silhouette Preprocessing	44
3.5	Bounding Box Localization of extracted Information	45
3.6	Silhouette Images	46
3.7	Grayscale Images	47
3.8	Image Mean	50
3.9	Eigenspectrum Representation	51
4.1	Within Class Regularization Scheme	58
4.2	Eigenspectrum obtained from the training data	61
4.3	Unregularized Eigenspectrum	65
4.4	Regularized Eigenspectrum	67
4.5	Different Eigenspectrum Modelling	68
5.1	Graphical Representation of Deep Learning Process	80
5.2	RELU Diagram	84
5.3	Sigmoid Diagram	85
5.4	Tanh Diagram	86

LIST OF FIGURES

5.5	Softmax Loss Graphical Representation	95
5.6	Center Loss Graphical Representation	96
5.7	Proposed Joint Supervisory Loss Graphical Representation	96
5.8	Weizman Images	104
5.9	KTH Images	104
5.10	Softmax Accuracy	104
5.11	Center Loss Accuracy	105
5.12	Proposed Model Accuracy	105
5.13	Dropout Visualization	107
5.14	Dropout Effect on KTH	115
5.15	Dropout Effect on Weizman	118
5.16	Loss function Graph	122
5.17	Accuracy Graph	123
5.18	Regularized Cost	124
5.19	Regularized Accuracy	124
5.20	Convergence Graph	130
5.21	Convergence Graph for 100	132
5.22	Convergence Graph for 200	133

List of Tables

2.1	Summary Point of Related Articles in Human Activity Recognition . . .	17
3.1	Confusion matrix of the recognition evaluation in % using Principal Component Analysis for silhouettes features extracted from the Weizmann database	54
3.2	Confusion matrix of the recognition evaluation in % using Principal Component Analysis for grayscale features extracted from the Weizmann database	54
3.3	Confusion matrix of the recognition evaluation in % using Fisher Linear Discriminant Analysis for silhouettes features extracted from the Weizmann database	55
3.4	Confusion matrix of the recognition evaluation in % using Fisher Linear Discriminant Analysis for grayscale features extracted from the Weizmann database	55
4.1	Confusion matrix of the recognition evaluation in % using Principal Component Analysis for different activities of the Weizmann database	72
4.2	Confusion matrix of the recognition evaluation in % using Linear Discriminant Analysis for different activities of the Weizmann database . .	72
4.3	Confusion matrix of the recognition evaluation in % using three parameter modeling for different activities of the Weizmann database	72
4.4	Confusion matrix of the recognition evaluation in % using four-parameter eigenfeature regularization and extraction from various activities of the Weizmann database	73

LIST OF TABLES

4.5	Confusion matrix of the recognition evaluation with our three parameter using the ANN classifier on the Weizmann database	73
4.6	Confusion matrix of the recognition evaluation with our four parameter using the ANN classifier on the Weizmann database	74
4.7	Confusion matrix of the recognition evaluation in % using Principal Component Analysis for different activities of the KTH database	74
4.8	Confusion matrix of the recognition evaluation in % using Linear Discriminant Analysis for different activities of the KTH database	75
4.9	Confusion matrix of the recognition evaluation in % using three-parameter modeling for different activities of the KTH database	75
4.10	Confusion matrix of the recognition evaluation in % using four-parameter eigenfeature regularization and extraction from various activities of the KTH database	75
5.1	Model Architecture	102
5.2	Accuracy score for different loss function method	106
5.3	Key training summary of input parameters to the CNNs	109
5.4	Model Architecture for convolutional neural networks with Weizmann dataset	112
5.5	A summary of model evaluation for CNNs using KTH datasets	113
5.6	Evaluation metrics with CNNs model for KTH dataset	114
5.7	Confusion matrix of the recognition evaluation in % using convolutional neural networks for different activities of the KTH database	114
5.8	A summary of model evaluation for CNNs using Weizmann datasets	117
5.9	Evaluation metrics with CNNs model with Weizmann dataset	119
5.10	Confusion matrix of the recognition evaluation in % using convolutional neural networks for different activities of the Weizmann database	119
5.11	Hyper-parameters summary and dimensional search range	128
5.12	Stochastic hyper-parameter selection for initial training	128
5.13	A summary of CNNs evaluation using stochastic default hyper-parameter with KTH datasets	129
5.14	Results associated with running hyper-parameters optimization on CNNs model for human activity recognition	131

1

Introduction

1.1 Background

The recognition of human activity has become a popular area in computer vision systems because of its numerous applications in the security sector, patients monitoring, human computer interaction and the gaming industry, to mention but a few. Recent advances in artificial intelligence have pushed the technological boundaries in all the computer vision sectors. In particular, there is an ever growing necessity and endeavours to improve automatic human activity recognition. A comprehensive feature engineering process is vital to the goal of improving computer vision systems, feature extraction and its engineering process are basic skills for crafting human activity recognition model. Computer vision is a branch of artificial intelligence that enables machines to mimic human vision. In other words, it aims to capacitate machines to make visual sense of their surroundings and thus take appropriate action(s) in the same way as humans do. Notably this field of study has received tremendous growth in recent times due to its potential application in almost every aspect of modern living [3, 99]. The proficiency of any vision system depends on the maneuverability of its pixel values, such that a coherent meaning and understanding of any scene can be recognized or classified [2]. Computer vision technology has found wider applications in industrial robotics, autonomous vehicles, object tracking, human activity recognition and face recognition. For the computer vision system to be good at making decisions like humans, a lot of feature engineering is needed for such system to be able to make adequate sense of its environment before informed decision is made. The art of crafting discriminative fea-

tures to promote adequate feature learning cannot be over emphasized. This is because discriminative features are vital for effective class labeling. This creates not only better classification, but also improves object recognition and allows complex body pose problem experienced in human activity recognition to be overcome. Learned features from any vision systems can proportionately be affected by various factors such as: dimensionality, environmental factors, high pixel correlation, scaling and model complexity. These problems in their very least form can prevent successful crafting of discriminative features. The ability to leverage on domain knowledge in order to proffer adequate solutions to these mitigating factors is also very key for building effective computer vision systems [1–4]. Intelligent cameras like the one shown in Figure 1.2 are equipped with softwares that are capable of infusing automation to video recognition systems. These softwares provide the intelligence that gives the cameras the ability to detect and classify numerous actions performed by humans or another object in an environment. Such camera systems have also found useful applications in robotics, video surveillance, intelligent automobile and military equipment. Recent global security challenges in most places of the world have allowed greater investment to be channeled towards the study of criminal tendency. The benefits of human activity recognition can range from the surveillance of public places such as bus stations, patient monitoring to more complex scenes such as crowd monitoring and border security. The dynamics of criminality, terrorism and other negative vices that are inimical to the society are tremendously evolving. Hence, this thesis has endeavored to proffer the state-of-the art feature regularization solution to help improve the intelligence of the computer vision machine. The areas where these technologies are deployed are often deemed to be critical and nothing is left to chance. Therefore, to obtain maximum functionality of the computer vision model, there is need to provide reliable and sustainable state-of-the-art technology. In recent times, the computer vision and imaging communities have been awash with different models for building intelligence for recognition. While a considerable amount of energy has been spent on this research, there still exist a huge gap in the feature learning process. For instance, considerable energy has been directed towards research in human activity recognition (HAR), human-computer interaction, entertainment, remote sensing, monitoring of elderly patients in care homes and hospitals[5–9]. The common challenges associated with human activity recognition (HAR) are the shape and posture representations, background clutter, image, viewpoint variation, excessive

illumination and statistical image representation [10–13]. These challenges unduly introduce excessive complexity. Therefore, the task of building a complex model to solve complex issues in the field of computer vision has its own disadvantages. Firstly, such a complex model is susceptible to overfitting the model, a situation where the model fails to generalize well with the test data even though they have good recognition score with training data. Secondly, their convergence time to optimal solution is generally too poor. The two challenges described are major issues that open the wider discussion of feature regularization to have great discriminative features.

Earlier research in Computer Vision [14, 15] has shown that panoptic surveillance largely depends on human activities recognition and in some cases facial recognition. The recent ubiquitous security challenges are enormous and the quest for robust security measures has become a key element in strategic policy implementation in most organizations, be it private, cooperate body or governmental institutions. Again, vulnerable patients in most hospitals largely depend on monitoring gadget that are built from human recognition technology. Gadgets such as intelligent cameras are deployed for the monitoring of real time of the elderly, weak and in most cases children with special needs. So, lots of advantages are gained when practices that improve the science of obtaining accurate and high recognition score are projected to new technological frontiers. The regularization of the features is an efficient method that can be used to improve model recognition, while better generalization in the model is also an advantage drawn from the regularization of features.

Feature extraction and its engineering process are important skills for building human activity recognition model. However, depending only on extracted the features for learning often leads to poor model generalization which is highly error prone when used for HAR [5]. In view of these limitations, this research work seeks to address such potential problem by introducing a feature regularization scheme. This feature regularization is used for encapsulating discriminative patterns that are needed for better and efficient model learning. Feature regularization creates means for model expressive characterization and recognition needed in most computer vision applications. Model complexity is one of the main reasons why representational and hand-crafted learning is difficult. The presence of excessive noise in the training data can introduce superficial complexities; the variances introduced by these complexities in the training set do not present useful information during the test phase. These dissimilarities in both training

and test set are the main culprit that generates overfitting in machine learning model. These additional superficial complexities which are adequately modeled in the training sets are not necessarily accounted for in the test sets. This imbalance remains a major challenge in computer vision [87].

The main research goal is to answer the question on "How to effectively improve the science of feature crafting and manipulation, through various regularization means for HAR". As shown in Figure 1.2, video surveillance is deployed for observing most neighbourhood. The reason for having video cameras range from crime detection and prevention, data collection and other scientific researches [2]. In the past, video cameras were passive in their function, humans or law enforcement agencies would only proceed to analyze recorded scenes if a heinous crime had taken place. Figure 1.1 shows a surveillance camera technology at its earliest stage, with a recording device attached to the camera. Humans spend thousands of hours watching and observing behavioural activities recorded in the storage device. The footage analysis of these stored video frames are constantly being searched by humans for any information. This kind of footage perusal often became tedious and unreliable because fatigue and complacency are very common in the analysis of passive video cameras. The development of intelligent algorithm that can leverage on best feature crafting in shallow or deep network of computer vision system is important for accurate representation and interpretation of human actions

Similarly, with the popular demand in the use of computer vision technology, areas such as automobile company, hospitals, prisons and public space surveillance departments have found great usefulness in computer vision algorithm. It has become critical to improve the techniques that are core in building such algorithms in machine models, and such improvements add to the general success in driving the frontiers of reliable detection and the recognition of human activities considered in this research.

Investment and large-scale research in deep learning architectural model has become very popular in the last decade. The concept of building shallow networks and creating handcrafted features for human activity recognition is receiving less popularity and this is because of a paradigm shift in the building of deep convolutional architecture that can extract and learn features in an end-to-end manner. Unlike the shallow architectures with simple layers capable of performing non-linear feature transformation, the deep layer architecture is more complex with the ability for learning convoluted non-linear

features. Therefore, as computer vision application processes evolve over the years, it has become expedient to develop cutting-edge state-of-the-art model that can increase the discriminative power of features from the deep learning model. Deep learning models are built to mimic the functionality of layers of neurons found in the neocortex, which is a part of the brain where higher-order functions such as sensory perception and all other cognitive processes are coordinated. The Deep learning model has become a variant subset of artificial neural network of choice for learning and recognizing digital, image and other data representation.

This study comprehensively explored the Convolutional Neural Networks (CNNs) and various means of features regularization, learning and building models that can efficiently recognize human activities or behavioural patterns. This kind of network has become the choice architecture because it is well suited for image and other object classification. Firstly, adopting this architecture enables quick and effective training of convolutional networks, thus providing the opportunity to learn directly from the images without the use of hand-crafted feature manipulation. Secondly, it also provides the ability for deeply training the many layers that may be present as this kind of deep training promotes better classification of images and learning hidden patterns present in the datasets.

In this study, an attempt was made to elucidate on hand-crafted and the deeply learned feature extraction methods has these two major ways of feature extraction are very common in computer vision. With the eigenspectrum regularization method, feature regularization is achieved through a meticulous effort of hand-designing features. Such features have shown to be very discriminative in the application of vision, [16–20] Notably, the tasks of engineering such features are too tedious, time-consuming and require domain knowledge. A proper feature engineering process in any machine learning task is much more reliable in its predictions and its ability to generalize. With the deeply learned features, its feature engineering process is almost always done through an automatic process. The features are extracted with the help of numerous cascades of filter layers, activation functions, pooling layers and the dense layer attached to the output. A combination of these processes is what is called the deep learning model which is used for classification and recognition purposes. Both methods of feature extraction were examined and the commonality between these two methods is that both can suffer from the curse of dimensionality, therefore they become susceptible

to the problems of overfitting and poor generalization in the machine learning model. By the regularization scheme proposed in this work, a high level representation of extracted features can be obtained, thereby putting the designed model at their best performances.

The Principal component analysis (PCA), linear discriminant analysis (LDA), fisher linear discriminant analysis (FLDA) and other non-linear kernel subspace methods like the kernel principal component analysis (KPCA), kernel linear fisher linear discriminant analysis (KLDA) have been presented in most literatures that attempted to proffer concrete solutions for the effective recognition of human action [10, 21]. However, the commonalities among these methods are their shortcomings, examples of which are the inability of these models to handle high dimensional data effectively and singularity issues arising from the sample size and ineffective feature extraction methods.

1.2 Research Motivation

The global use of video information in virtually every technological sector has necessitated the incorporation of intelligent algorithm in most computer vision systems. In [22], it was projected that the analysis of activity (parsing temporal sequences of object observations to produce high-level description of agent actions and multi agent interactions) amongst similar detections, tracking and the analysis of human motion will be the most researched in the nearest future. The explosion of research in the area of human activity recognition in the last decades has not only confirmed this prediction but significant progress has also been achieved in the videos and scene analyses. The ubiquitous deployment of the video recording machine for strategic surveillance reasons (military intelligence, law enforcement and commercial purposes), is a pointer to the proposal of Lee et al. [22]. These are sectors where adequate response and intelligence gathering are key for preventive and proactive purposes. The art of crafting and learning different methods of extracting high representational features is vital in reaching the goals of building the state-of-the-art recognition systems. Therefore, the main objective of this study was to develop efficient feature regularization methods that can improve the growth, accuracy and effective generalization of computer vision models. In this thesis, the computer vision subfield of human activity recognition is concentrated on.

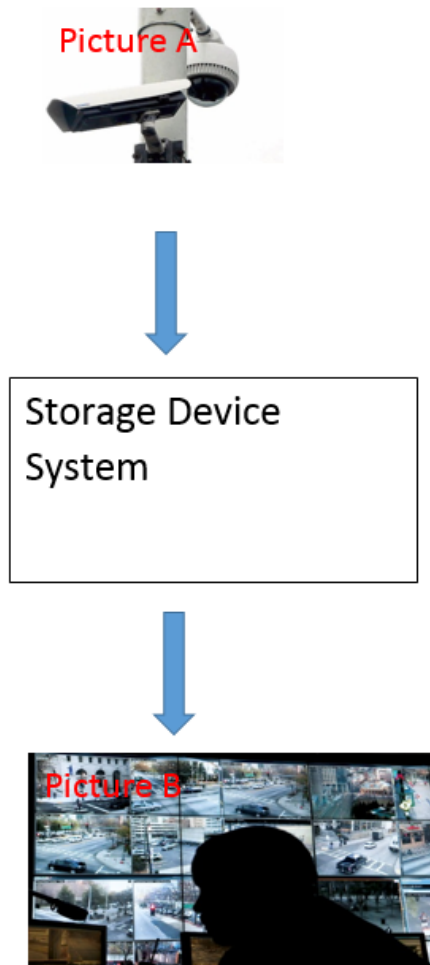


Figure 1.1: Passive Cameras - Diagrammatic Representation of a Surveillance System Using Traditional passive Cameras

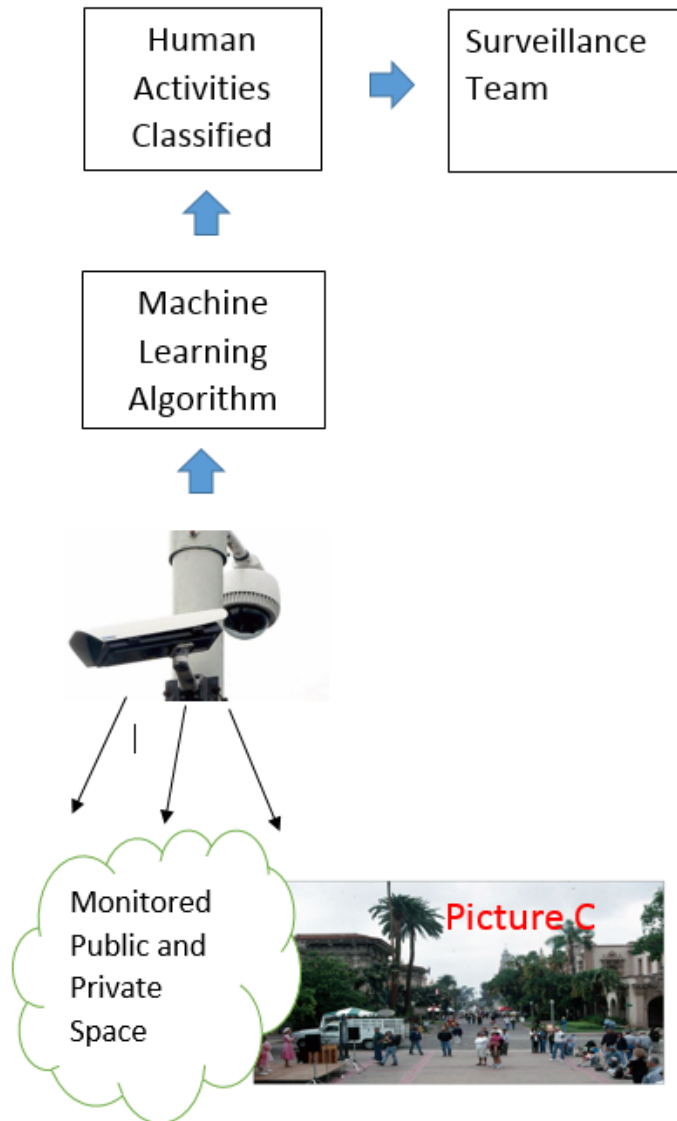


Figure 1.2: Intelligent Camera - Diagrammatic Representation of a Surveillance System Using Intelligent Cameras

1.2.1 The Main goal of the study

This study aimed to improve the recognition and classification of human action using our proposed improved regularization method. A new eigenspectrum regularization modelling scheme was developed to handle hand-crafted features. To realize these noble objectives of building an effective learning model that can discriminate human actions and interpret these actions from a scene, this thesis showcases the use of eigenspace regularization to achieve better state-of-the-art results. The likelihood prior-probability regularization is introduced to regularize jointly supervised loss function of deeply learned features. Deeply learned features are better and effective in feature discrimination. This study therefore limited its feature learning and regularization process to enhance human activity recognition.

1.2.2 Specific Objectives

A summary of the objectives for this study were as follows:

1. To create intelligent algorithm for human activity recognition purpose.
2. To develop a four-parameter model for effective eigenfeature regularization to enhance classification and recognition in HAR.
3. To proffer better ways of feature extraction.
4. To show the effect of applying the likelihood prior-probability regularization in deep learning for extracting quality feature for classification purposes.
5. To investigate the effect of within-class matrix on the recognition model.

1.3 Thesis Overview

Six chapters were used to relate and discuss this thesis. The chapters highlight the contents, depth and empirical findings while undertaking this study. The chapters are structured to reflect progression and advances in computer vision and image processing fields.

Chapter One focuses on the generality of the subject matter describing the recognition of human activity and the importance of feature learning and regularization.

Chapter one also highlights the introduction, research motivation, aim of the study and contribution to knowledge.

Chapter two describes a chronological narrative on previous studies that have been done in human activity recognition and diverse methods and approaches engaged. The traditional hand-crafted and the deep learning methods of feature extraction were reviewed extensively to identify current challenges and possible solutions.

Chapter three provides a window that enables a concrete understanding of the concept of image information detection and preprocessing of such images. This chapter also underscores the need for image preprocessing to allow for accurate representation of information that will allow the formulation or adoption of a better recognition model.

Chapter four discusses and investigates the effect of singularity problem associated with the within-class matrix in traditional linear discriminant analysis. The regularization of the within-class-matrix using a 4-parameter constant was extensively analyzed and results evaluated against state-of-the-art method.

Chapter five relates the concept of using deep learning model Convolutional Neural Networks (CNNs), for representational learning of human activity recognition. The CNNs hierarchical network model was robustly analyzed with each layer telling a unique story about every feature map extracted. Additionally, chapter five discusses the likelihood regularization parameters. Chapter five also delves on how different hyper-parameters can be used for the regularization and optimization of the recognition model.

Chapter six concludes the study. An indepth review of the contributions made towards the research are highlighted and possible future research area annotated for further exploration.

1.4 Contributions

In this thesis, the study and its findings have contributed to knowledge as follows:

- The within-class matrix has significant influence on the accuracy of a model designed for human activity recognition using the subspace method. This work has demonstrated that such accuracy can be achieved by reconstruction of the true variances in the data by a unique method of modelling the eigenspectrum by a 4-parameter constant.

- To underscore the relevance of feature scaling, this work highlighted the concept and principles of using scaling to improve feature extraction. This concept of scaling has shown better recognition when implemented for behavioural recognition and human action classification.
- The hierarchical nature of learning pattern representation from videos and still images of human activity are explicitly covered using the deep learning method. First, the use of Convolutional Neural Network has seemingly demonstrated its powerful and better feature extraction techniques. Secondly, a good concept of feature regularization showcased in deep learning loss function has demonstrated its uniqueness in improving recognition in HAR.

2

Literature Review

The complexities that surround recognition and classification in the field of machine learning are enormous. Challenging as it may be, this area of research has received diverse and dedicated attention from various computer vision and machine learning groups in recent years. Automated video surveillance (group or individual), computer-human interaction, remote sensing, image retrieval, entertainment and monitoring of the weak and vulnerable in homes and hospitals are some of the applications of computer vision and machine learning [5–9]. Recent global security challenges have also necessitated the incorporation of intelligent surveillance in major public places in the communities that we live in. However, the onerous task of having human resources to analyze this video sequence can be very expensive and prone to human error occasioned by human fatigue and lack of concentration. Secondly, these video recording machines are passive in action (lack proactive ability to prevent abnormal behaviour and activity classification). Consider the non-action sequences in the video frames, this could be boring for human interpretation. These disadvantages had fuelled the desire for effective and real time video interpretation for efficient classification and recognition purposes. Holistic subspace approach has evolved over the years so much that significant results have been achieved in human recognition and classification purposes.

Past developments in activity recognition have created lots of theoretical and empirical platforms that have given researchers many opportunities to explore. In the recognition task, many believe these tasks are decomposed into three major sub-tasks, motion analysis of human body parts, high-level tracking of human motion with multiple camera or single camera and the use of tracking features extracted from motion

sequences for human activity recognition. In [2], a review on interpreting human motion was analyzed using these three major parts for recognizing human activity. However, while a lot of energy was focused on shape and geometrical characteristics for obtaining meaningful recognition results, little was said on how discriminative features would be extracted and processed for quick recognition and classification purposes. The use of both 2-D and 3-D shape characteristics is analogous to the contour and volumetric approach in capturing basic human shape orientation. The 2-D method was conceived to deal with modeling of human action with or without explicit shape. The characteristics and addition of the volumetric approach in the human body shape have given better description and details of the human body in a 3-D. This has better tracking of human motion [2, 23]. Object tracking is essential in behaviour recognition, but a quick analysis and correct classification of such objects and their actions add importance and colour to the tracking activity. Shape and motion-based classification have been used as methods for object classification. Lun Zhang [24], with different camera views formulated a method that performs real time object classification and this was achieved by applying appearance-based techniques. The multi block local binary pattern (MB-LBP) technique which is excellent for its ability to encode rectangular regions and label different image structure is used to encode the appearance of objects. With the enormous overhead associated with this method, it has become almost impracticable for real time classification to be true in automated feature learning and HAR. Therefore, Hota, [25], proposed a better method to overcome the limitation of MB-LBP and the method uses an online feature selection method. A subset of these online features is aggregated to be learned by a machine model for classification and object recognition.

While most of the aforementioned techniques concentrated on the use of feature set for the classification of two or more classes[24], Gurwicz, [26] proposed a multi-class object classification in both low and high-resolution images for real-time video surveillance architecture. In this method, several morphological, textural, temporal, and periodical features were used to boost the discriminative power of this multi-class classification model. Another contributory factor to the success of this scheme was the abundance of dataset that were available for learning, making unconnected features which were less significant to become meaningful and contributive for image classification. In general, the task of action recognition can be grouped into three main areas and they are dimension reduction, feature extraction and classification of human

actions. These three steps are key for effective recognition after the detection stage has been finalized. Pixel values of images present high correlation and as such present multi-collinearity that affect the performance of the machine learning model. The collection and processing of millions of these kinds of highly correlated pixel can be very large and therefore difficult in terms of computational resources and this is a primary cause of poor classification in HAR. Therefore, most machine learning models have invested in data reduction techniques to actualize better recognition and classification of images in the computer vision field. The Principal component analysis (PCA), canonical correlation analysis (CCA), linear discriminant analysis, (LDA), have been used by most researchers for dimensionality reduction of data [7, 27–30] and the results show that a significant progress for data reduction was achieved. Feature extraction deals with the process of obtaining categories of features that are capable of describing expressively the meaningful information necessary for analysis and classification. In [31], feature extraction was grouped into two main categories and they are the local descriptor features and the holistic features. The methodology of the local descriptor leverage on the neighbourhood information to describe salient points associated with local motion rather than the shapes, while holistic features are used to capture shape features and other motion energy of images. The combination of hybrid features type for human activity recognition was performed by Sun et. al [31], the 2D and 3D SIFT local descriptors were combined with holistic feature obtained from the Zernike moments (single frame and motion energy image). The final combination of these two hybrid features shows that feature fusion has the capability to increase the performance of action recognition model. Wei et. al [32] investigated a much higher hybrid feature weight fusion for HAR purpose. In their work, a higher recognition rate was achieved when three descriptors were fused together to gain better recognition of HAR. Three feature combinations were clustered together, they are three-dimensional histogram of oriented gradient (HOG3D), a global descriptor based on frequency filtering (FDF) and the local descriptors based on spatial temporal interest points (STIP). PCA was then used for dimensionality reduction before support vector machine (SVM) and a multi class classifier was, therefore, used to establish the effective exploration of multi- class action. Again in [33], it was stated that curvelet transform technique offers unique directional and edge representation features. Therefore, in their work, a fusion of edge, texture and silhouette shape eigenfeatures were combined for the purpose of

the recognition purpose. A unified framework presented by Dinesh and Kuldeep [34], developed a technique based on the spatial distribution of the gradient (SDG) at several decomposition stages. This framework seeks to relate computational outcome of SDG on average energy silhouette images (AESIs) and the AESIs is used for estimating shapes and action in HAR. For effective description of AESIs, spatial distribution of gradients at different stages and the sum of the directional pixels (SDPs) deviations were calculated. The calculation of the temporal makeup of the shape information of the silhouettes was done by applying R-transform. Firstly, R-transform was extensively used in [27] for solving the problem of continuous distance change of an object as it relates directly with the size and posture of image character. Two viewpoints presented by two cameras of unequal distance can potentially result into having same image represented differently, this can become a major disadvantage in the representation and extraction of features. Therefore, the R-transform which can extract periodic, scale and translation invariant features is a better option for resolving different viewpoints and difficulties. Secondly, a nonlinear subspace method Kernel discriminant analysis (KDA) is introduced for extracting unique features that can discriminate similar activities. The training and recognition was done by applying the hidden markov model (HMM) classifier, and a 95.8 percent was recorded. The recognition of human activities in video was conducted in [35], and emphasis was centered on accurate and distinctive extraction of information feature vectors. These vectors were collected by the introduction of Directive Local Binary Pattern (DLBP) features. These kinds of features were more superlative in their orientational information than information captured by directional magnitude when handling binary silhouette recognition. The projected Directive Local Binary Pattern (DLBP) integrates coordination information with intensity variances of binary shape images. It is further pooled with Edge Orientation Histogram (EOH), producing better and unique feature set which can help in image and object recognition. The training and recognition were done by a support vector machine (SVM) and it was reported that this method can improve the limitations inherent in the Local Binary Pattern (LBP). Sabanadesan et al [36] developed limited feature-based activity images. The solid HOG (Histogram of oriented Gradients) and histogram of resemblance forms are mined from videos. Numerous forms of support vector machines were applied to construct distinct codebook for all activity class as compared to alternate single class. Every input feature is then illustrated by Locality constrained Linear Coding (LCC) in

conjunction with codebook essentials and the combination of spatio-temporal features. This method enables a robust construction of the dictionary before SVM is applied for classification. Zhang et al [37] consolidated the conventional clustering method by developing the subspace clustering method for improving the difficulty experienced by the conventional clustering method inability to handle multidimensional dataset. These techniques make use of the density-based clustering method that is capable of obtaining clusters by utilizing axis-parallel subspace. Data collection is achieved by the sensor's devices and features necessary for human activity recognition are gotten and clustered. The non-parametric similarity in the trajectory data motion feature in relation to the Bayesian network proposed by Neil et al [38] was used for the recognition of human activities. A tracker that functions well with colours is then used for data collection and the Hidden Markov Model is used for inferring different activities on the dataset [5, 7, 39]. Examples of these subspace methods are the principal component analysis (PCA), discriminant analysis (LDA) and the fisher linear discriminant analysis are the most common models that have gained popularity in the subspace method of recognition and classification of human actions. Other non-linear subspace methods that have also gained valuable recognition in computer vision sector are the kernel variants such as the kernel principal components analysis (KPCA) the kernel fisher linear discriminant analysis (KLDA) and these have also been applauded for their individual advances in human activity recognition [10, 21]. A summary point of some work done by previous researchers using similar subspace methods and other techniques have also been tabulated in Table (2.1) to give a clear direction on the human activity research work.

These subspace models became the focus of attention due to their successful results shown in face recognition [29, 67]. Hence, their extension to a more complex nature as human activity recognition. The PCA is a handy method that captures the variances associated with a data set and this variance is most significant along the principal axis. The uncorrelated principal components are drawn from the correlated data by projecting the entire data as orthogonal transformation to a much lower subspace and by this action, the variances in the data are then made explicit. These explicit principal components can point out the basic direction and differences in a data. The principal components in the newer lower subspace are able to preserve much of the variances in the data by a set of eigenvectors with the highest eigenvalues [68, 69]. While the

Table 2.1: Summary Point of Related Articles in Human Activity Recognition

Main Work/ Contribution	First Author,Year
Group recognition	[40] T. Lan
Human action recognition in videos using long short term memory	[41] M. Baccoude, 2010
Automated human action learning via combination power of CNN and RNN	[42] M. Baccoude, 2011
Generative model for learning object recognition	[43] J. Susskind, 2011
Image recognition and denoising using deep Boltzmann network	[44] Y.Tang, 2012
Heirarchical modeling of human activity recognition	[45] T.Lan, 2012
Automatic recognition of human action in surveillance video	[46] S.Ji, 2012
Image classicication with deep convolutional network	[47] A.Krizzhevsky, 2012
Action recognition in video using deep convolutional network	[48] K.Simonyan,2014
Historical overview of deep learning	[49] J. Schimidhuber, 2014
Local spatio-temporal information analysis using deep learning network	[50] A. Karpathy, 2014
Depth of convolutional network and recognition accuracy investigation in image analysis	[51] K.Simonyan, 2014
Human activity recognition using deep learning	[52] H.Iang,2014
Deep sparse autoencoder for human activity recognition	[53], H. Liu, 2014
Shape modeling with deep network	[54] S.A. Eslami
Group and individual recognition in surveillance scene with deep learning	[55] Z. Deng, 2015
A two satge hierarchical model for group activity recognition	[56] M.Ibrahim, 2015
Efficient utilization of deep neural network model for image classification	[57] C. Szegedy, 2015
Human action modeling in video sequence using deep network learning	[58] J.Donahue,2015
Deep hybrid features learning for activity recognition	[59] M.Hasan, 2015
Human pose labeling with deep neural network	[60] K.Fragkiadaki,2015
Pixel-level labeling with deep learning in activity recognition	[61] S.Zheng, 2015
Human motiom modeling via spatio-temporal and deep learning	[62] A. Jain, 2015
Group action recognition using recurrent neural network	[63] Z.Deng, 2016
Depth map application with convolutional neural network for human activity recognition	[64] P.Wang, 2016
Human motion analysis with deep metrics learning	[65] X.Yin,2016
Deep recursive and hierarchical modeling of human activity recognition	[66] T.Liu, 2016

principal components are vital and significant for retaining a major part of the original data, the non-principal components are assumed infinitesimal and are often discarded [29]. Therefore, discarding the eigenvectors that retain the non-principal components eigenvalues amount to losses in the PCA model and this is the primary cause of its poor recognition ability. Again, another disadvantage of the PCA method is the weakness in encoding class label information thereby making it a poor model for classification. The LDA model is another subspace method that has also gained relevance in human activity recognition. This method attempts to locate the best axis in the subspace which maximizes the distance between the label of dissimilar activities while minimizing the distance of similar activity. Encoding labels is a unique characteristic of LDA as this feature makes it a better technique for handling labeled datasets unlike its PCA subspace counterpart. The between class matrix accounts for the maximization of the variances between dissimilar activities, while the within-class matrix minimizes the variation of similar class. In the works of [29, 70, 71], LDA was reported to be more efficient than the PCA and that this was due to its feature extracting power. However, because of the problem of singularity common to high dimensional datasets like human activity dataset, this method suffers from the inability to have inverted within-class matrix. Taking the inverse of the within-class matrix is often too problematic because the covariance estimates cannot attain a full ranked status and, therefore, cannot be inverted. Fishers proposed the Fisher's linear discriminant analysis (FLDA) to solve the singularity problem caused by small sample size on LDA and his idea was to use the PCA to reduce data dimension before performing LDA. At this stage, the bulk of the principal components are retained, and the non-principal components are discarded before the LDA method is implemented [5]. The application of PCA at the onset for data reduction seems to have solved the singularity problem. However, the trade-off was the loss of information with the PCA application at the initial stage. This trade-off has undermined the success recorded with the FLDA method. Another variant of the subspace method is the DLDA. This method has its within-class scatter matrix null space discarded while retaining eigenvectors with the least eigenvalues [72, 73]. The disadvantage of this method is the undue scaling characteristics obtained when smaller eigenvalues are used to scale their corresponding eigenvectors and this can affect the recognition results. Bappaditya et.al [5] offered a holistic technique of eigenspectrum regularization to enhance the recognition method using three parameter constants. A

2.1 A Brief Historical Review on Deep Learning

breakdown of the within-class eigenspace into three different sections is done and these sections are regularized from each other. The succinct and sure start and end limit of these three subspaces area can be very hard and challenging to determine. The three subspaces region formed by subdividing the complete eigenspace can also be too burdensome and susceptible to error. For that reason, one of our research questions was centered on how to advance the latest state of the art performance chronicled in [5]. A modeling of the within-class matrix using additional parameter was a conceivable idea and this work has produced a better feature extraction and representation. The variances related to the within-class matrix of our model was effective in capturing key information on the dataset and this has led to terrific discriminative power without having to discard useful eigenvectors as practiced in most subspace methods [49, 57, 74, 75].

2.1 A Brief Historical Review on Deep Learning

Until recently, many computer vision and machine learning techniques have been using simple non-hierarchical shallowed architecture for pattern recognition and classification purposes and tremendous success has been recorded with the use of these techniques. A notable characteristic of these shallow architectures contains a simple layer of non-linear feature transformation model which is a sharp contrast to its deep counterpart with much complex form of convoluted non-linear features. An example of commonly used shallow architectures are the hidden Markov models (HMM), conditional random fields (CRFs), support vector machine (SVM), Gaussian mixture models (GMMs), linear and non-linear dynamical systems, multi-layer perceptron (MLP) with a single hidden layer, maximum entropy (MaxEnt) models and Kernel regression. The uniqueness of these shallow architectures is their relative computational stage simplicity and typical arrangement with a single layer of transformation. This simple layer can map raw input from their immediate environment to a more separable feature sub-space than their original space. The SVM and other known conventional kernel methods illustrate this simple shallow architecture as its linear separation pattern is strengthened with a one or zero feature transformation layer often done with the help of the kernel tricks. While shallow architecture has demonstrated its strong capability of solving simple related problem in most areas of machine learning (simple object recognition), such

2.1 A Brief Historical Review on Deep Learning

problem-solving skills become inadequate when more complex environments (natural language processing, computer vision system, human speech recognition, signal processing and so on) are to be modeled. Figure 2.1 illustrate different human actions ranging from simple action performed by individuals to more complex action undertaken by more persons. Designing the representation abstraction power is a function of each scene complexity and these complex scenes cannot be handled by shallow architectures. Hence, this millennium has embraced the use of deep representational power of neural networks often referred to as deep learning in most literature[76]. This kind of learning has recently become prominent in extracting complex features and enables multiple level representation and having feature vector reuse ability. For example, a lot of computer vision systems are increasingly being designed with layered hierarchical structure that provides quick and real time pixels information processing in complex scenes thereby providing adequate in-depth interpretation of various actions in such environments[64, 77, 78]. Deep learning architecture are designed to have numerous layers of non-linear processing agents where the input to a layer above is an immediate output from a layer beneath. For the purpose of classification, such higher layer of representations is used to develop key and important aspects of the input to aid discrimination, while suppressing less discriminative and redundant parts of the input. The great success achieved in the field of deep learning are partly because of their unique generative nature, also with the discriminative ability the additional layers offer. Secondly, the unsupervised pretraining procedure inherent in most deep learning architecture allows a huge number of unlabeled training data for the effective representation of structures and hidden patterns in the input features[75, 79]. Historically, the ubiquitous relevance of deep learning originated from artificial neural networks and it has witnessed a lot of transformational development over the years. Such models with concatenated non-linear layers of neurons have been in existence since the 1960s [80–83]. Models that exhibit this deep architecture are the feed forward neural network, MLPs with numerous latent layer and other deep model variants recently developed (see Section 2.2 and 2.3). Back-propagation, a well known weight adjustment method for NNs improvement and refining tricks have been in existence for long [84–86]. However, it gained greater popularity in the 1980s. Fortunately, back-propagation method was no longer a panacea for accurate results in NNs as researchers quickly learnt about its numerous demerits in learning networks with more and complex hidden layers (

2.2 Building Deep Representational Learning

Review works of [87] and [88] attest to this. Back-propagation minimizes errors through local gradient descent that initializes its start point randomly from any given point. With a greater depth of neural network connection, the error propagated get barreled in the local optima often called the local optima problem. This leads to a common error that causes gradient exponential shrinkage or growth often referred to as vanishing or exploding gradient respectively. The challenges posed by this gradient inefficiency has compelled researchers to walk away from using backpropagation for neural networks weight adjustments. More recently, researchers have turned to more efficient shallow models (SVNs CRFs and MaxEnt model) for obtaining robust global optima together with convex loss function. Nowadays, most researchers proceed first by training their models on shallow problems and then try to extrapolate their solutions to fit a deeper model. The introduction of unsupervised learning algorithms has helped steer the optimization of deep models to a positive direction and this was fully discussed and highlighted in [89, 90]. This kind of model is often generative and building such deep models for optimization purpose has received great attention in recent times[16–20].

2.2 Building Deep Representational Learning

Geoff Hinton in 2006, expanded representational learning popularly referred to as deep learning in recent times. This representational learning is the primitive feature of learning variants method that are useful in the holistic description of data structure. The importance of depth in representational learning underscores the need to highlight different forms of building deep representational learning (see Section 2.3). The composite and heavy computational nature of network depth makes it more difficult for training purposes and a lot of research has been directed towards solving this particular problem in the computer vision sector [65, 76, 91–93].

Deep learning presents two major advantages to computer vision. Firstly, feature reuse in deep learning has been a formidable point that explains the strength inherent in distributed representations and theoretical gains recorded in multiple levels of representation or hierarchical feature learning. Secondly, deep architectures are capable of learning abstract features at higher layers of representation in a progressive form.

2.2 Building Deep Representational Learning

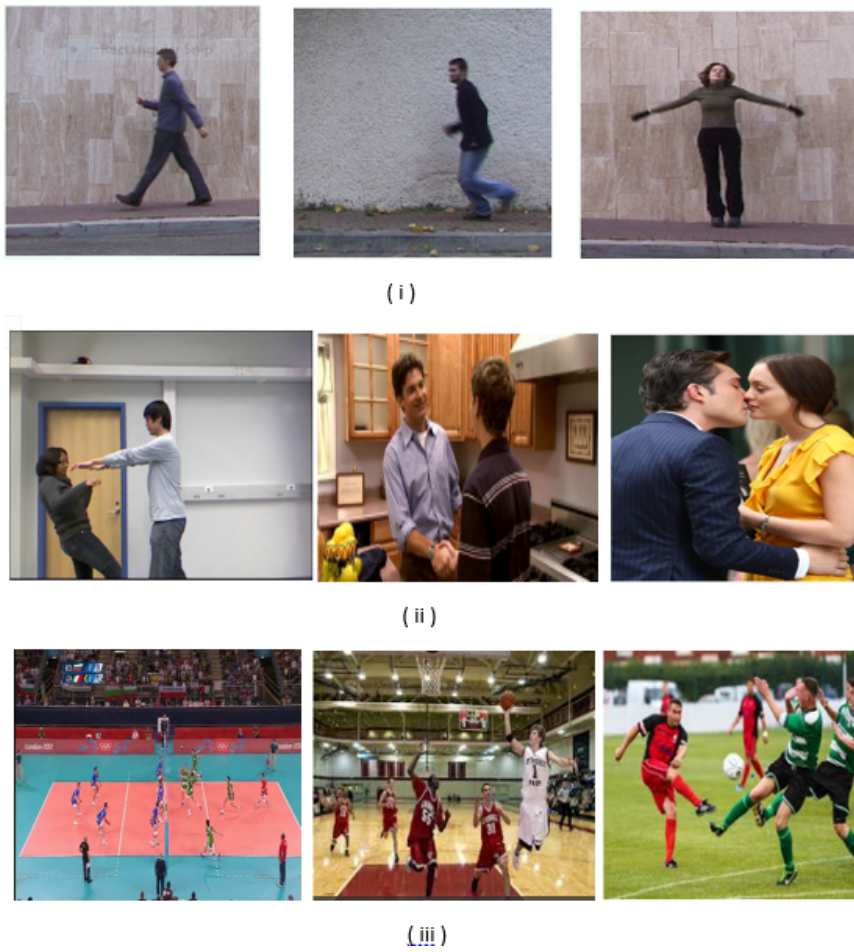


Figure 2.1: Different Human Action - (i) Single person action (ii) Two person interaction (iii) Group activity action.

2.2 Building Deep Representational Learning

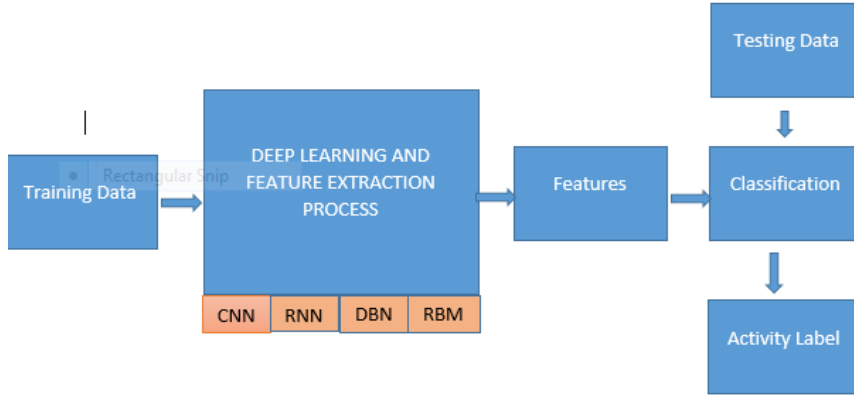


Figure 2.2: Feature Extraction and Training - Block Representation of Deep Learning Model

Theoretical results in [94–99], show clear directions of progress with deep learning representation efficiency than the shallow or insufficiently deep network model. The block illustration of deep learning stages as shown in Figure 2.2 is also like the traditional hand-crafted method of feature extraction except that the algorithm and techniques inherent in the black box are different CNN steps with multiple processing units for better feature extraction. In this chapter, the highlight of evolving practices in deep learning is discussed and particularly recent gains in human activity recognition. A significant milestone has been recorded in predicting various human activities, even in hierarchical complex environments such as group recognition have also benefited a great deal. While every effort has been made by different authors to present a better representational model capable of learning and exploring human actions and their primitives dynamics, there still exist a huge gap that needs to be bridged in the recognition and classification tasks. Such obvious gaps in the computer vision sector underscores the need to build robust and deep representational models.

Several research studies[89, 100, 101] were put forward by researchers in same year and many others have taken tremendous interest in deep learning including authors of [88, 102]. Learning of stacked features at one level at a time which was referred to as the greedy layer wise unsupervised pretraining is a phenomenon that uses unsupervised

2.2 Building Deep Representational Learning

feature learning for specific learning purpose at each level. The features learned at this level are used as input to subsequent immediate upper levels. A resultant effect of this action places additional layers of weights to deep neural networks. Furthermore, the combination of this unsupervised layer can then be used to initialize deep supervised neural networks. A generative model that reflects such initialization of deep supervised network is the deep Boltzmann machine (DBM) (Subsection 2.3.1). In [103, 104], it was observed that stacking of extracted features in a layer wise form is a veritable tool in harnessing deep and better representation of hard to model salient structure that improves recognition with less classification error and leverage on the quality of sample built from probabilistic model [105], or improving the invariance characteristics of the extracted feature [106]. The deep features obtained from the greedy layerwise unsupervised pretraining can be used to initialize a deep supervised neural network or as input to a purely supervised machine learning model like the support vector machine (SVM) [107]. An attempt has been made in combining layers of pretrained unsupervised learning so that an excellent and efficient unsupervised model can be created. However, while different methods have been proposed, there has not been a clear distinction as to which method performs the most. The earliest work done was stacking pretrained RBMs to form deep belief network [89] or DBN, in which the concept of Bernoulli-Bernoulli RBM learning of binary features creates an input from the probabilistic activation of the hidden layer. Thus, the output from one RBM machine acts as an input to another RBM layer. A first-hand implementation of the layer-by-layer greedy learning in [52] shows that this strategy is indeed efficient and helpful in human recognition and other classification tasks. Secondly, the autoencoder is another veritable model capable of building deep learning networks (Subsection 2.3). This method of deep learning attempts to model the system input to become the output by adjusting the weights of the hidden layers to perfect the encoding scheme. This is well suited for video streams because of its robustness in hierarchical distributed feature leaning method used. Also, it is well suited for modeling high dimensional time series events like human motions [53, 65]. Different methods of back-propagation types (for example conjugate gradient method, steepest descent, and so on) techniques are used for training the autoencoder. However, it has been observed that the back propagating error in a large deep network can become very tractable and this is a major disadvantage of this method. Another approach that has been proposed is the combination of RBM

2.2 Building Deep Representational Learning

parameters into the deep Boltzmann machine (DBM). The DBM is another known member of the Boltzmann machine family having its unit like the RBM arranged in layers. Unlike the RBM, its layers are of multiple hidden units. The goal is to half the RBM weights such that the DBM weight can be obtained. The likelihood maximization unlike the RBM is not done directly. Instead the lower bound likelihood is maximized by some chosen parameters [108].

2.2.1 Related Works in Group Activity Recognition Using Deep Neural Network

The explosion of research in activity recognition of human behaviour has led to a further incursion into an in-depth study of human interactions with scene, object, and their spatial or temporal relationships with others (see Figure 2.1). The latter of these type of human interaction is commonly known as group activities. The concept of reasoning about structures plays an important role in group activity recognition, effective representation and interpretation of group activity is aggregated from the dynamics of the individual persons in the group activity. The spatio-temporal description between individual and the scene is very vital to group activity recognition. Mostafa et al. [56] proposed a deep hierarchical architecture to model group activities using a structured temporal framework. A two-stage approach is used to implement this model. The temporal representation of information in the scene from both stages is based on the LSTM model. The first stage of their proposed method is modelling individual person temporal dynamics and secondly combining the person level information from the first model for inferring group activities. This model was reported to be successful in video surveillance, video retrieval and in sport analysis[56]. Considering the fact that competitive results can be achieved in human activity recognition by variants of deep neural networks, better classification results can be obtained when deep neural networks are combined with a probabilistic graphical model as conducted by Z. Deng et al[55]. The work of Deng et al.[55], focused on the combination of CNNs with the graphical model. The introduction of the graphical models as reported in [40, 45] allows hierarchical structures to be integrated into group activities modelling and other human interaction for the purpose of recognition and classification. The key components of the probabilistic graphical model are derived from the multi-step message passing neural network and the label adjustment is done via belief propagation layers of the neural network.

2.2 Building Deep Representational Learning

Again, Deng et al.[63] in a similar combination of graphical and deep neural network was able to bridge the difficulty that exists in the use of low level concept output in interpreting high level compositional scenes. The use of a generative architecture (Recurrent Neural Network) is used instead of using other traditional inference methods. Antic and Ommer [79] proposed the use of semi-local parts (latent constituent) in interpreting group activities. This method is based on learning the classifier with parts that are functionally related which in general are the group activity constituents. Lan et al.[109] proposed three different approaches (adaptive structure learning, feature level learning and the combination of both learning methods) for finding the interactions between low level person-person relationship to a higher group-level interaction. The adaptive latent structure learning was modeled to reduce the redundant person actions that are not relevant in group activity modeling and recognition. Thus, this model dynamically decides inter-relationship between group members. Donahue et al. [58] proposed a "doubly deep" model (spatially and temporally deep) capable of handling computer vision sequential interaction dynamics. In this model, a deep hierarchical like structure (CNN) in combination with LSTM for training a recognition model is developed. The model can encode temporal state dependencies. The doubly deep model function by transforming visual input (video frame) to a fixed length feature vectors with the help of convolutional neural network, the output from the CNN becomes the input to the second non-linear model(LSTM) that is connected to an output for recognition and classification. The Network-in-Network proposed by Lin et al.[110] is used to leverage the representational power of neural networks by increasing the convolutional depth with a 1×1 convolution. While 1×1 convolution increases the depth, a better classification is reported. Szegedy et al. [57] introduced deep neural network architecture that is like that proposed by Lin et al. [110]. This model was able to capture representational power in their architecture, both in depth and width using the 1×1 convolution building blocks. The most remarkable aspect of this method is that it enhances the utilization of the computing assets of the network. Mohamed et al. [111] developed a deep graphical model that is able to represent long-range and higher-order spatiotemporal dependencies of video features using a robust inference model called Hierarchical Random Field (HiRF). The (HiRF) concept combines input features to form mid-level video representations which are reflective on its performances as a model that can be used for effective recognition and localization in group activity

2.3 Different Classes of Deep Learning architecture

modeling. Unique characteristics of this model are its abilities to perform collective foreground grouping while discarding features estimated as background noise. The Fusion of convolutional network proposed in [50] came in handy for its unique form of fusing different input layer frames with different deep networks. The combination of two streamed CNNs as proposed by [48] highlighted the effectiveness of combining a CNN with a priori RGB frames and another CNN trained with a pile of 10 flow frames. The state-of-the-art results recorded in the shallow model of HDMB [112] and UCF101 [113] is comparatively like the combinational averaging effect using both RGB and flow frames. Furthermore, the use of the Support Vector Machine (SVM) for fusing RGB and flow methods has improved the recognition results as opposed to the simple score aggregate as earlier mentioned. Like [56], however, with a 3D convolutional neural network, deep spatio-temporal features are also learned [46] and [42]. In [42] and [41], a comprehensive study is done on visual and motion features extraction and the use of recurrent network for shaping temporal dependencies.

2.3 Different Classes of Deep Learning architecture

Deep learning as earlier discussed, refers to the hierarchical composition of non-linear information processing paradigm. The techniques and architectural model intended for use are influenced by the complexity of recognition/classification task. This section captures an overview on different deep learning architecture and they can be categorized into two major classes (Generative and Discriminative).

2.3.1 Generative architecture

Generative deep architecture is a robust approach that is used for achieving high-order correlation features of observed data in most machine learning tasks. In this type of architecture, the process of feature learning is unsupervised since the labels for the data are never used for learning purposes. By so doing, the entire data hidden structure is learnt dynamically. The advantage of generative architecture is its ability to aid supervised learning by its "pre-training" module. This enables the lower levels of deep neural networks to be trained layer by layer (bottom-up) before committing to the overall learning of the entire layer of the deep network. With this method, the architecture (generative) is best whenever there are small samples of training data.

2.3 Different Classes of Deep Learning architecture

The most prominent subclasses of generative deep architecture are the energy-based deep models and an example of such is the autoencoder and others extensively discussed in this chapter ([3, 53, 59, 65, 114, 115]). This generative model has not only demonstrated its usefulness in dimensionality reduction [90] in image processing and computer vision field, but also has aided efficient recognition and classification of human activity action [53]. The deep autoencoder is a variant of the deep neural network whose output mirror image the input data, thus encoding the hidden features of its input and attempts to generalize its output data. Again, its rich data dimensionality reduction property has contributed to its fame in computer vision. Simply defined, it is a non-linear feature extraction method that has no class labeling. It comprises of an input layer, a smaller hidden layer(s) and an output layer which is a replica of the input layer. The autoencoder is considered deep when it has one or more hidden layer units. The autoencoder's ability to fine tune and incrementally update its parameters, continuous learning of models of scene and activity is possible with dynamic environment [116, 117], when new unlabeled data are available for activity representation or recognition. In [65], temporal alignment is a vital preprocessing stage for the recognition of human action. Metric learning is the main challenge common to the temporal alignment of labeled time series. The authors in [65], used a non-linear metric learning (deep autoencoder) to obtain the spatio-temporal features for effective metric-based comparison and recognition of human activity (HAR). In [53], Liu and Taniguchi used a sparse autoencoder for the extraction of low level features. These features are employed to realize high generalization performance. With this method, quick and concise representation of human motion is achieved. Another outstanding generative model that has also found usefulness in the area of deep learning is the deep Boltzmann machine (DBM) [118–120]. DBM are Markov Random Fields composed of multiple layers of latent variables whose layers are interconnected while the latent variables that are in each layer are interconnected and are separate from the layers above. Each layer in DBM is capable of capturing complex and higher-order correlation in the data and internal representation of input data is well learned and this is beneficial for the recognition purpose. A concise and accurate representation can be built from a huge available unlabeled input while fine-tuning the model with a labeled data is possible for the task at hand. The restricted Boltzmann Machine (RBM) which is a building block for DBM is formed when the hidden layer is constrained to be one. Stacked

2.3 Different Classes of Deep Learning architecture

layers of RBM improve better features learning as the output of one RBM represents the input to another RBM above and such layers result in the Deep Belief Network (DBN). In recent times, RBM has found useful application in speech [121] and facial expression [43] and this has also extended to include using the RBM method in dealing with occlusions and noise removal in complex scenes using multiplicative gating [44]. The peculiar characteristics of the successful modeling of spatial[54] or temporal [122] patterns in most dataset has given RBM its unique colouration of feature extraction. Hence, in [123], the generative RBM model (Local interaction RBM) is used to extract spatio-temporal patterns in high dimensional data and these extracted patterns have proven to be outstanding in multi-class human activity recognition. Another form of deep generative network is the Recurrent Neural Network (RNN) which has been used to model and generate sequential data [62, 124]. The number of layers in RNN can be representatives of the input sequence length. This method is widely recognized due to its powerful ability to handle sequences of data (speech, text and videos sequence) and in recent times has found usefulness in the area of human activity recognition [125–127]. RNN has demonstrated its learning prowess in notable end-to-end learning tasks [58, 60, 61, 128]. However, the major setback of this techniques has been the difficulties associated with training its model which, a consequence of its vanishing gradient characteristics. Spatio-temporal graphs are veritable and powerful tools used for representing high-level spatio-temporal structures [4, 129–134]. In [62], the authors emphasized the importance of combining such spatio-temporal graphs with the RNNs sequence learning capability to produce more valuable recognition models for human actions and their activities. The proposed approach experimented on (human pose modelling, human-object interaction and driver decision making) shows great success over other state of the art approaches in computer vision.

2.3.2 Discriminative architecture

Discriminative architecture uses techniques of conditional probability distribution on input data as it relates to the corresponding output label. This concept has also found recognition in shallow architecture like the Hidden Markov Model (HMM)[135, 136] or the Conditional Random Field (CRF) [137, 138]. The CFR is graphically an undirected model that conceptualizes the conditional probability relations that exist between an

2.3 Different Classes of Deep Learning architecture

output label given a sequence of input data. The success of this model is in its ability to capture dependencies that are obviously hidden between various activities. The non-deterministic nature of human activities (diverse pose orientation) has made the CRF algorithm more suitable for determining the dependent relationships in such data sequences[1]. In recent times, deep-structured CRFs have seen immense development for deep learning purposes. The deep-structured CRFs is realized by combining outputs from the lower layer of the CRF model with input data and making these two combinations a representation of its higher layer[139]. Various variants of the deep-structured CRFs have found useful applications in the area of human activity recognition[66], electronic recognition[140], natural language processing[139] and identification of spoken language [141]. In [66], the limitation of CRF was the model inability to completely capture the intermediate structure within the target state. While only few time step interactions are represented, higher order dependencies necessary for modeling most complex applications are unavailable. In view of these reasons, the authors proposed a more effective technique using the deep recursive and hierarchical conditional random fields (DR-HCRFs) to model the intermediate representations contained in the targets (human-object relationship) [142, 143]. The capturing and representation of a deep-order temporal dependencies in the data shows that a combination of CRFs and other deep conditional probability variants are richer with contextual information. Another prominent deep and discriminative architecture is the convolutional neural network, which has different module with each containing a convolutional layer and pooling layer. The convolution helps create filtered feature maps that are stacked up on top of another. Weight sharing is a common feature seen in convolutional layer while sub-sampling of convolutional layers is made possible by using the pooling layer. The pooling phenomenon helps reduce the dimension of intermediate representations aiding the reduction of computation from subsequent intermediate lower layers. However, the downside of this may include the loss of information on the data structure. Again, translation invariance is achieved when a max-pooling layer is cascaded with a convolutional layer. In [144], it was argued that representing such invariance for convoluted recognition tasks is difficult to achieve and, therefore, advocating for a better way of advancing and handling invariance in dataset. Nevertheless, this method has been very successful and it has found useful applications in the area of computer vision and image recognition[19, 47, 115, 145, 146]. In [46], 3D convolutions in convolutional

2.4 Deep model Training and its challenges

layers of CNNs was used in extracting spatial and temporal dimension features in human activity recognition. This technique allows rich and important motion information to be present in all adjacent frames. This provides different channels of information from a single input, thus giving this method its performance advantage because all the final features obtained are representations from different channels. The computational layer (a computational block for CNNs called DaConv, which can be regarded as a convolutional layer endowed with the ability to adapt the scale of the filter kernels), with computational power of CNNs has been used seamless in representing depth information in most RGB channels for the deep representation of human and robotics recognition reasons [147].

2.4 Deep model Training and its challenges

The combination of the single-layer model resulting in deep layered network has extensively been discussed in previous sections. However, not much has been presented about the challenges of training deep model for recognitions and classification purposes. In this section, more attention is dedicated to how joint training of these layers could be achieved and also highlight some of the challenges that can possibly be encountered. The concept of nonlinearity in deep learning is often associated with the higher level of abstraction needed in utilizing efficient modeling of human activity recognition. Such a higher level of abstraction often becomes too sensitive in dealing with the complex nature of human posture representation considering that a slight change in human posture may be interpreted differently. Therefore, presenting manifold human poses to such a model may become a complex phase as mapping input to representational space is very challenging. Thus learning a robust representation technique that unfolds the input manifold of human posture has been known to improve the training problems common in HAR [148, 149]. The concept of training deep architecture has been around since 2006 (Section 2.2) even though convolutional networks training has been ubiquitous for a much longer period[145]. Supervised or unsupervised layer wise training has influenced recent pattern recognition learning. The use of unsupervised pretraining to facilitates supervised learning was aimed at directing the effective training of transitional representations, and this intermediate illustration enables effective

2.4 Deep model Training and its challenges

piecewise learning of features which may be difficult to learn at a go. A corroboration of this pattern is also reported in the curriculum learning [150] which was highly suggestive of the simpler training concept first before the higher levels of layers can be composed. Priors learnt from intermediate representations in deep learning have shown to be successful in semi-supervised embedding [151]. The reasoning around the importance and how unsupervised pre-training was used to catalyze the deep learning process is extensively researched [104] and researcher's intent was to find a pathway that would effectively create regularization and optimization effects. Regularization is key in the training of deep network model to achieve robust generalization, and avoid over-fitting and thus improve performance. The dropout approach has been widely reported to be effective [152], even though a regularization method called swap-node was subsequently introduced. In [104], the prominent effect of regularization was even more obvious in the use of stacked RBM and deep auto-encoders for the initialization of supervised classification neural networks. The use of unsupervised learning for fine tuning the learning dynamics and for proper initialization has helped in reducing generalization errors. The hypothesis of this method underscores the importance of the features derived from such regularization because these features inherently capture the principal variation both in the input distribution and output targets of interest. A well-known problem of deep neural network is the optimization of the lower level in reference to a supervised training. This can be very difficult because the top two layers of the deep network cause over-fitting to the training set irrespective of what kind of features (good or bad) that were computed. A slight variance on the numerical pre-sets of the optimization measure can have a huge effect on the joint training of deep architecture and the non-linearity concept used and initialization boundaries deployed are also key factors [87]. One important premise upon which the difficulty of optimization of deep architecture is derived is from the singular value of the Jacobian matrix used for feature transformation from one level to another level. The propagated gradient tends to diminish as it travels down the layer if the singular values are very small, allowing contractive mapping in all directions. Resulting from the difficulty experienced during optimization, the necessity of seeking second-order methods for the purpose of optimizing deep architecture and recurrent networks was investigated by researchers [153]. In 2011, the Hessian-free second order optimization was used to alleviate the basic deep

2.4 Deep model Training and its challenges

learning challenges in RNNs. This was indeed most superior to the standard gradient-based LSTM RNNs on so many tasks [17, 18]. It has also been reported that the use of other RNN algorithms seldom give better performance than steepest descent for LSTM RNNs[154–157]. In [158], recurrent neural network and RBMs training are done by unsupervised pre-training enabling the utilization of good features captured in variable states. The use of the natural gradient[159] methods has proved to generalized well in networks with very many parameters as was proposed by Pascanu and Bengio [160] Roux et al.[161]. An adaptive learning rate devised in training RBM was proposed by Cho et al. [162]. The whole idea of this method was to have a gradient estimator who keeps track of invariance resulting from flipped hidden bits and detect inverting signs which are analogous to its weight vector. The practice of initialization as reported in [156] has shown that keeping the Jacobian of each layer to approach unity in all its singular values is a panacea for circumventing the difficulty of deep training. Sutskever [16], in his empirical work in training deep architecture of recurrent network, acknowledged the importance of the guided initialization procedure and the success achieved. In the deep learning architecture, hidden or latent nonlinear units are strong objects that are very core to both training and generalization performance. The experimental results of [16, 163, 164] shows how much such influence these nonlinear hidden units can have on its overall developmental training process. Again, the use of sparse rectifying units has also shown to be a robust form of improving quick data convergence and generalization [47, 163–165]. Other efforts made at biasing neural network training were to invalidate the average cost and ensuring that the slope of each hidden unit outputs is conditioned to have a null value [166] and making a normalized magnitude with the local boundaries [165]. Finally, the concept of layerwise training has been set aside. This is because recent research findings show that optimizing the initialization process and a careful selection of the nonlinear unit of very deep supervised network are trainable without relying on the layerwise concept of pre-training [16, 47, 167, 168]. Researchers have argued that given such conditions (optimized initialization and concise selection of nonlinear unit of large dataset), layerwise unsupervised pre-training offers no better advantages over purely supervised learning when given the necessary time of training.

2.5 Summary and Discussion

This chapter outlined the recent advances in human activity recognition using both hand-crafted and deep learning method which is a brief historical chronology on subspace and deep learning was also analyzed in this review highlighting continuous evolution and underscoring the importance of artificial neural network in machine and computational learning. The generative and discriminative classes were used as a categorization scheme to further buttress some of the prominent deep learning architecture in literatures. There exists numerous literature on deep learning, particularly from the machine learning community. However, the study has been restricted to focus mainly on the human activity recognition perspective, recognizing the fact that more work has been done in this area in the past seven years as can be seen from Table 2.1 This chapter has not only discussed both hand-crafted techniques and two major categorization schemes in deep learning, it also serves as a veritable window for the proper understanding of in-depth analysis of complex feature vector representation, learning, optimization and classification in HAR with high level representational learning. This chapter thus provided better understanding for researchers who want to expand their scopes on feature learning and optimization in HAR methods and how they are better utilized under different circumstances and scenes. Again, the fact that different methods are unique in their feature extraction process, the combination of one or more deep learning methods showcased in this chapter is a pointer to the malleability of deep learning methods. They can be adjustable and shaped into any desired form with various algorithms. The key information tailored has been on building, learning and understanding variants of robust deep learning architectures and other hand-crafted methods with emphasis on structured hierarchical development of human activity features. The discussion was on the gains and the ease of learning parameters in a piecewise mode and challenges that could occur when such learnings are initiated at a goal. There exists other channels by which the discussed methods in this chapter can be improved by having a full understanding that better feature regularization and learning deep has a lot of potential in solving complex learning problems in human activity recognition. Understanding has influenced this research decision to develop better feature regularization method using parameter selection and learning deep features that are salient with hand-crafted

method. The remaining chapter of this thesis explores better methods of feature regularization and learning deep to attain the better recognition model. Furthermore, in this review section, it has been established that the success derived from deep learning cannot be misconstrued to have one solution to all classification problems. On the contrary, each and every technique used is unique and depends on different problem formulations. What generalizes well in one problem formulation may likely and inherently fail in another. Therefore, this research is advocating that researchers seek more ways of inter-operating deep learning methods and other machine learning so that a broader solution is achieved. An open question yet to be answered is how hierarchical graphical models can be intertwined with other known kernel methods for classification purposes. Finally, the importance of optimizing deep learning techniques was discussed. The local optima problem resulting in vanishing or exploding gradients and how they can be leveraged are key elements of successful implementation of deep learning. In view of the foregoing, the theoretical know how of feature regularization in the subspace method and deep learning should be encouraged to help solve myriad of complex issues in the computer vision community. Effective feature building and representation among others have shown to be invaluable in building robust classification and recognition models. This can be achieved with a well-planned feature learning architectural algorithm to attain state of the art results for human activity recognition purposes.

3

Image Detection, Extraction and Pre-processing

3.1 Introduction

Consider that numbers of frames in videos are directly proportional to the time of recording, the quick and effective analysis of such video frames has become even more difficult because of many inactivity that can intermittently be observed when the object of interest is not visible. For any meaningful recognition model to be built, it is expedient that the object of interest be first detected. A proper representation of video information is an important phase that enables a robust video analysis and the extraction of features. The information of interest in the case of HAR will be the human actions that are partly or fully visible, and this constitutes the video information that is required for evaluation as it relates to understanding the relationship between environmental scenes and human interaction. The detection of human activities in videos is significantly different from still object detection. Therefore, moving regions are thought to be very crucial in video analysis. Detecting such moving information is one of such important phases. Some methods used for detecting human activities are the gaussian mixture model, background subtraction and optical flow method. Our detection model that was used in this work is the background subtraction [169, 170]. Furthermore, most of these images detected are prone to a lot of problems that could be affected by background clutter, extreme illumination, object variation induced by viewpoints, shape, posture, environmental and hardware noises [10–13]. Hence there

is a need to ensure that the detected images are preprocessed in other to mitigate the effect that could be caused by the factors earlier listed above. Some of the crucial steps for preprocessing are image filtering, grouping connected components with a similar size threshold, and performing other morphological operations like hole filling.

3.2 Image detection method

In background modeling, the mixture of Gaussians' approach has been a common method used for object localization and detection in video frame recorded by a stationary camera. Stauffer and Grimson [170] established the novel concept of modeling background for the purpose of learning and obtaining foreground detection. Friedman and Russel [171], extended background modeling to areas of traffic surveillance. In their work, a mixture of three Gaussians was used for modeling individual background pixels; each of the three Gaussians modeled are the road, vehicle and the shadows respectively. The estimation algorithm is used to reset the model for the initialization purpose after which an empirical process is used to label the opaquest component as the shadow, while the two other components are labeled as the vehicle and road respectively with the most outstanding variance being considered as the vehicle. The foreground detection process involves a comparison of the individual pixels with each Gaussians and the values obtained in respect of the Gaussians help to classify which of the pixels is the foreground. To preserve this process in the real time mode, an incremental estimation maximization algorithm is used to update the learning parameters. Having a better knowledge of the colour history of each pixels $\{X_1, \dots, X_t\}$ by a mixture of K Gaussians was modeled and facilitated by the work of Stauffer and Grimson [170].

Primarily, the intensity of each pixel can be described in an RGB colour space. Considering the current pixel value, the probability of observing each pixel is formulated and given in 3.1.

$$P(X_t) = \sum_{i=1}^K \omega_{i,t} \cdot \eta(X_t, u_{i,t} \sum i, t) \quad (3.1)$$

For every pixel in the background, they are all considered to be a mixture of K Gaussians. The Initialization of different parameters that make up the mixture of Gaussians is first considered as soon as the background model definition has been completed. Where K represent a given number of distributions, $\omega_{i,t}$ is an approximation of weight

3.2 Image detection method

of the i th Gaussian mixture at time t , $u_{i,t}$ explains the mean value of the i th Gaussian in the mixture t , and the covariance matrix of the i th Gaussian mixture at a given time t is represented as $\Sigma_{i,t}$; and η represents the Gaussian probability density function.

$$\eta(X_t, u, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t-u)\Sigma^{-1}(X_t-u)} \quad (3.2)$$

K is heuristically chosen in sympathy to the availability of memory and its values range from 3 to 5. Also, for reasons inclusive of computation, Stauffer and Grimson [170] assumed that the red, green and blue pixel values are independent of each other, but characterized by similar variances. Although, this assumption may not be the true reflection of their hypothesis, it has however, helped them evade costly matrix inversion in lieu of the model's accuracy. The representation of the covariance matrix is given in 3.3:

$$\Sigma_{k,t} = \sigma_k^2 \quad (3.3)$$

With the onset initialization of the parameters and the detection of the first foreground, parameter update is crucial for further detection as the foreground will most likely be moving. A criterion ratio $r_j = \frac{\omega_j}{\sigma_j}$ supported by the Stauffer and Grimson [170] can be used for the methodical ordering of the K Gaussians. An assumption of this ordering is that the background pixels with an extraordinary weight, but having a fragile variance is more likely to be the background because of its stationary character. Unlike other moving objects whose value vary constantly. The first B Gaussian distributions which surpass the definite verge T are then collected and kept as the background distributions, while the other distributions are foreground distributions in Equation 3.4

$$B = \underset{b}{\operatorname{argmin}} \left(\sum_{i=1}^b \omega_i, t > T \right) \quad (3.4)$$

A match assessment for each pixel is done for every new frames that come in at time $t+1$. If a pixel value equals the Gaussian distribution, it therefore means that the Mahalanobis distance as shown in equation 3.5 has its K value equivalent to 2.5. This condition results into two distinct circumstances, one of such circumstance being that the pixel value matches one of the K Gaussians.

$$\text{sqrt}((X_{t+1} - \mu_i, t)^T \cdot \sum_{i,t}^{-1} \cdot (X_{t+1} - \mu_i, t)) < K\sigma_{i,t} \quad (3.5)$$

In such circumstance, if the Gaussian distribution is recognized as part of the background, then the pixel is categorized as the background or otherwise as the foreground. On the alternative, failure to get a match between the pixel and the K Gaussians results in the pixel values being classified as the foreground. This allows for a binary mask to be acquired and an update of the parameters is done for further acquisition of subsequent foreground. A comprehensive revision of the importance and their settings were critically evaluated and discussed in [172, 173]. Considering the match test in equation 3.5, the foreground detection mechanism and the update distribution parameters are dependent on a match or a no match of the K Gaussians. When a match is made, the update process is shown in 3.6, 3.7, 3.8 as follows:

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} + \alpha \quad (3.6)$$

where α is a constant learning rate

$$\mu_{i,t+1} = (1 - \rho)\mu_{i,t} + \rho X_{t+1} \quad (3.7)$$

$$\sigma_{i,t+1}^2 = (1 - \rho)\sigma_{i,t}^2 + \rho(X_{t+1} - \mu_{i,t+1}) \cdot (X_{t+1} - \mu_{i,t+1})^T \quad (3.8)$$

where $\rho = \alpha \cdot \eta(X_{t+1}, \mu_i \sum_i)$ The weight of the unmatched component is substituted by $\omega_{j,t+1} = (1 - \alpha)\omega_{j,t}$, while η and \sum are unchanged.

However, in the event that a no match between the pixel and K Gaussians is the case, the least probable distribution is substituted with a distribution having the present value as its aggregate which is a value of a primary high variance and small prior weight.

3.2.1 Background Subtraction

Background subtraction is one of the many steps used for detecting video information (images). This method of image detection is best deployed when a still background is involved. The video segmentation of each frame is done to enable the subtraction between each frame and the background frames possible. The subtraction phase entails subtracting each frame from the background image so as to get the motion information

3.2 Image detection method

which is seen in the form of the silhouette [174]. To determine the probability that a particular pixel is part of a background, the Gaussian probability density function as described in (22) is used for the modelling of this process.

$$P(I(x, y)) = \frac{1}{b} \sum_{i=1}^b \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(\frac{-(I(x, y) - J_i(x, y))^2}{2\sigma^2}\right), \quad (3.9)$$

Where $P(I(x, y))$ represents the probability of a pixel background in the present frame, σ denotes the variation characterizing the intensity values and $I(x, y)$ represents the current frame intensity and $J_i(x, y)$ is the intensity value of the pixel appropriated at point (x, y) of the i^{th} background image. A transformation of the current frame to a binarized image is made possible with the help of a referenced threshold value that creates the contrast and equation 3.10 shows the binarized image.

$$B(x, y) = \begin{cases} 1, & P(I(x, y)) \leq th \\ 0, & P(I(x, y)) > th \end{cases} \quad (3.10)$$

The background and foreground understanding are necessary and helpful in motion information extraction. A reference threshold value often helps in a clear distinction between these two types of pixels. A pixels intensity that is greater than the threshold th value in its current frames is the background pixel, while those that are less than such threshold th are regarded as the foreground. The silhouette image that is produced from the background subtraction is shown in Figure 3.1. A detailed binarized extraction pseudo-code is presented in algorithm 1.

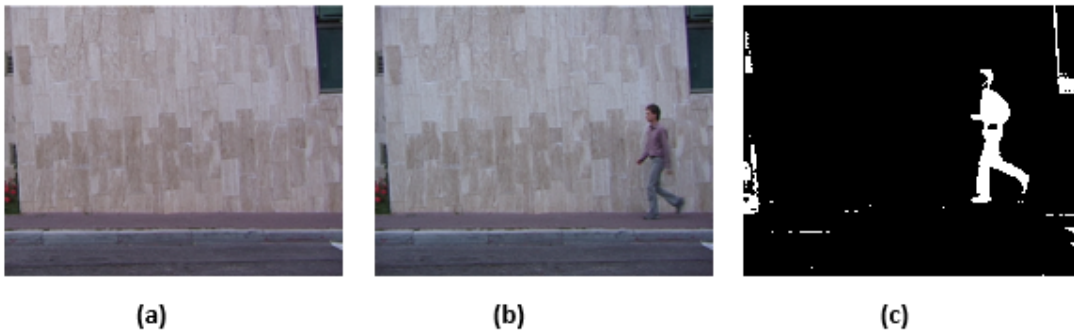


Figure 3.1: Background Subtraction for Video Information Recovery -
 (i)Background Image (ii)Information (iii)Recovered Information

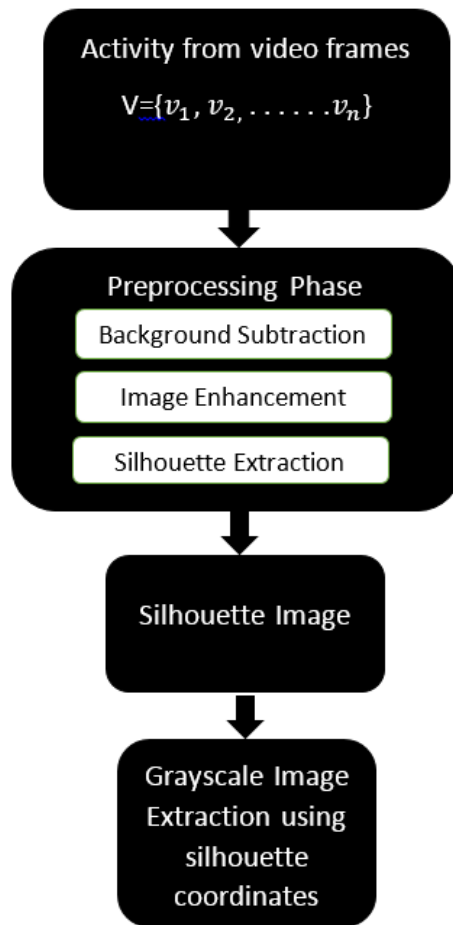


Figure 3.2: Video Information Extraction - Model for Image Extraction From Video Sequence

3.3 The Image Enhancement Process

Algorithm 1 Silhouette Extraction Pseudo-Code

```
1: procedure MYPROCEDURE
2:    $V = \{v_1, v_2, \dots, n\}$  VideoFrames
3:    $B = \{v_b\}$  Background Images
4:    $T =$  Preset Threshold
5:   For  $V = 1 \rightarrow n$ 
6:      $I = \{v_n\}$  Read in  $n_{th}$  image
7:     Frame Diference =  $abs((I) - (B))$ 
8:     if FrameDiference >  $T$  then
9:       return Pixel value is Forground.
10:    else
11:      Pixel Value = Background
12:    Binary Image stored
13:  End
```

3.3 The Image Enhancement Process

Noise, light variances and camera quality contribute immensely to the quality of video images. For example, excessive or less adequate lighting system can adversely affect the quality of binary state of the foreground or motion information. Again, environmental noises are also veritable factors which can create undue imbalance on the video quality. Therefore, the image enhancement process is a necessary option that cannot be overlooked. Such enhancements can alleviate and reduce to the barest minimum the undesirable effect of these external factors. While there are many ways of enhancing video images, this study only applied three major methods of enhancement that are immediately necessary to have quality images. The three processes of image enhancement applied were image filtering, retaining of connected components (areas that are larger than a set values are only retained) and the filling up of holes present in information object. In Figure 3.2, different stages that are needed to have clear and informative dataset are presented and each stage is dependent on the first previous layer above them. Image filtration and the smoothening process are the first stages of image enhancement to be explored and this allows for the video images to be disconnected from external environmental factors such as noise and undue lighting process. The filtration allows for stable and more robust video frames to be used in most activity recognition

3.3 The Image Enhancement Process

sectors. While some undesirable elements are filtered and smoothed out from the video images, others are still visibly clear. In the case of a background subtraction, care should be taken not to erode off motion information which is useful. This is a good reason why the filter kernel used for filtering video information may vary from one to another and they are also carefully chosen.

The concept of concentrating on the motion information which is vital in the entire process is schemed through the connected components method. This method allows areas larger than a set value to be retained while discarding areas that are less than a defined heuristically threshold value size. Furthermore, holes and other superficial patches resulting from other preprocessing stages are refilled to have complete and near perfect silhouettes images. Figure 3.3 shows the preprocessing stages after the background subtraction has been done. A noisy but binarized image from a walking video sequences is represented in Figure 3.4(a). The binarized image is filtered using the median filter. Thus, Figure 3.4(b) represents the filtered frame. Figure 3.4(c) depicts the connected components application and this causes the connected components that are less than a pre-set value to be removed. Figure 3.4(d) is a clear silhouette image after filling of the holes is done. The preprocessing done has the advantages of not only producing clear and quality background images, but is also very instrumental in the automatic cropping and extraction of 3 dimensional images from the parent video stream. This process is made possible by tracking and recording the silhouettes coordinate which is then transferred to the original video frame for the 3-dimensional image extraction.

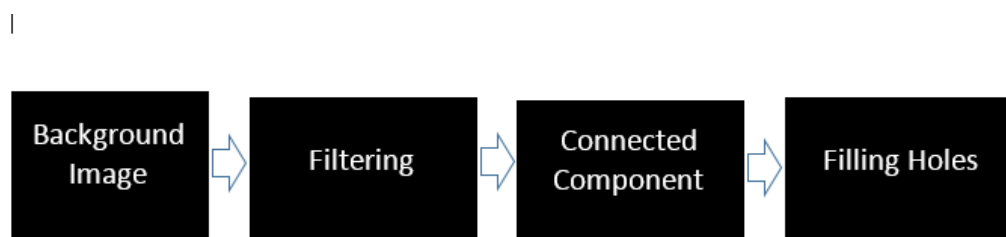
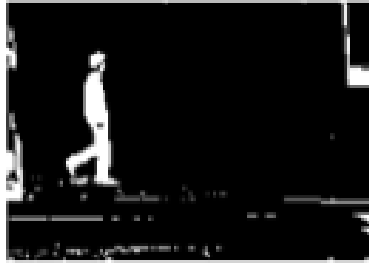
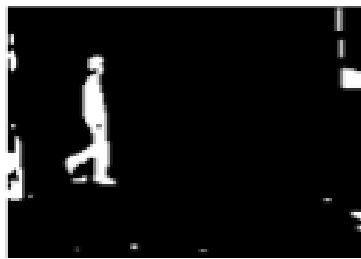


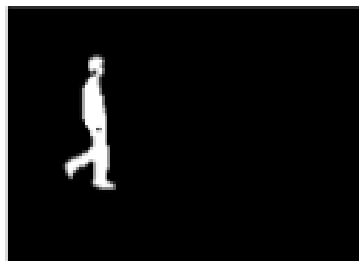
Figure 3.3: Image Preprocessing - The Preprocessing Stages of A Noisy Background Image.



(a)



(b)



(c)



(d)

Figure 3.4: Graphical Illustration of Silhouette Preprocessing - Preprocessing Steps for A Background Subtracted Images (a) Shows A Noisy (b) Filtered (c) Connected Component Application (d) Hole Filling

3.3.1 Extracting Silhouettes Images

The silhouette extraction system from a binarized image is shown in Figure 3.5 and 3.6. The silhouette extraction process requires that a motion information be identified and this identification can be done by setting the connected component values such that the area size specified in the background images are only retained. With such retention, the centroid of the connected components that represent our motion image are calculated and a bounding box is drawn over as it is shown in Figure 3.5. Again, using the coordinates of the bounding box, the area within such bounding box perimeter is extracted from the frame. This subset of the video frames contains the motion information (silhouettes). Firstly, a major advantage of this method is its uniqueness in reducing the dimensionality of video images, thereby reducing the computational burden on the model intended to use the dataset. Secondly, each of the silhouette coordinates can be transferred to the original video images for quick video image extraction.



Figure 3.5: Bounding Box Localization of extracted Information - Human Detection And Localization With A Bounding Box

3.3.2 Extracting Grayscale Images

The extraction of grayscale images for human activity recognition (HAR) is discussed in this subsection. For grayscale images, filtration and smoothening is also applied to produce better and clearer video images. Firstly, while the silhouette extraction is done directly from the binarized images, the grayscale images that were used were automatically cropped from their original images using the same bounding box coordinates on the silhouettes. As such, a one to one coordinate correspondence between



(a)



(b)

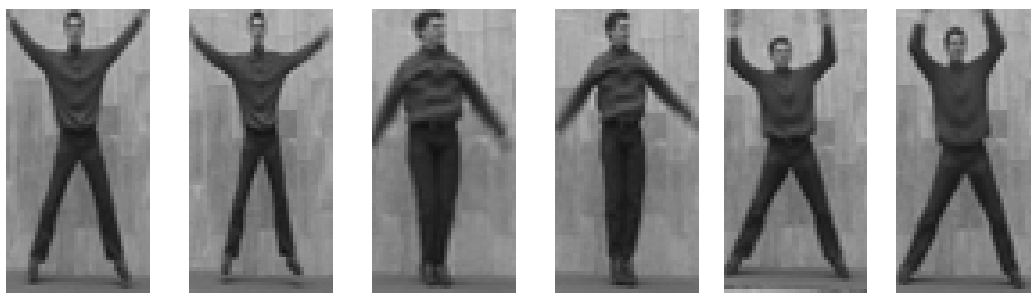
Figure 3.6: Silhouette Images - Extracted Silhouette From Video Sequences of (a) Bending (b) Side walking

3.4 Silhouettes Versus Grayscale Images

formed silhouettes and its equivalent original grayscale image is established. Secondly, the images could be cropped manually and labeled accordingly.



(a)



(b)

Figure 3.7: Grayscale Images - Extracted Grayscale Features From Video Sequences of (a) Running (b) Jacking

With the KTH datasets, each grayscale image is manually cropped and a histogram equalization is then applied to the cropped images. Figure 3.7 shows cropped grayscale of human activity Images performing running and jacking activities.

3.4 Silhouettes Versus Grayscale Images

Extraction of features from silhouette and grayscale images has been around for a long time in the computer vision sector. Deciding which of the two to use depends on environmental circumstances which is a concept that bothers on computational efficiency or type of model to be adopted for recognition purposes. In [174–176], low level sil-

3.5 Feature Extraction and Dimensionality Reduction

houette features were used as input feature vectors to generate the joint self- similarity for action recognition. In [27], the author's argument for using binary silhouette was based on the privacy of elderly people. While some model perform well with silhouettes images, others thrive with grayscale or coloured images. In Human Activity Recognition (HAR), the orientation of the human body is an important discriminative power and this can be captured by the body shapes. Shapes are best represented by silhouettes which can provide all the useful information regarding human body orientation [176, 177]. The shape information is clearly distinct from the background because of the binary state of motion information obtained after background processing.

The grayscale is a range of monochromic shades of gray with no colour and the darkest possible shade is black and this represents the absence of transmitted light while the lightest possible shades is white. Grayscale features have found wider applications in face recognition process [178]. However the extension of such application in HAR has greatly been acknowledged and evidenced by the amount of research publications in this area. In [5], grayscale images are used, and their features are extracted for HAR recognition purposes and experimental results show that the grayscales features obtained were also discriminative. Again, texture-based recognition and classification are generally biased towards grayscale and RGB images. Unlike the silhouette images, the grayscale and the RGB image pixel informations have a higher probability of sharing some similarity with the environment and this can become a real challenge. Deciding which of the two-image type to use can be a function of many factors, and these factors may range from environmental circumstance a which is concept that bothers on computational efficiency or the type of model to be adopted for recognition purpose.

3.5 Feature Extraction and Dimensionality Reduction

In this section, a quick overview on dimensionality reduction and feature extraction on both image types (Grayscale and Silhouette) are discussed. A major concern and challenges common with HAR is the fact that thousands or millions of pixels could be involved when building a classification model. The correlation between millions of pixel values from different images is the main challenge responsible for poor recognition in the HAR model. Furthermore, the computational burden of such model can be very exhaustive, thereby resulting in difficulty in building real time HAR project.

3.5 Feature Extraction and Dimensionality Reduction

Considering the foregoing, it is necessary to implement some sort of dimensionality reduction to enhance the extraction of relevant features that are important for classification purposes. For brevity, the Principal Component Analysis (PCA) and the Linear Discriminant Analysis (LDA) are highlighted to showcase their feature extraction and dimensionality reduction ability.

3.5.1 Principal Component Analysis

This method is known for its second order statistical-based analysis capable of encoding global information on mean faces or eigenfaces in face recognition. In HAR, they are important tools for representing flexible unit of the body. PCA is a popular method for transforming original data to a lower dimensional space [179]. The original data is mapped into a smaller subspace through a linear combination of the top eigenvectors and the uniqueness of this method is the ability for the original features to be preserved in the newly formed smaller subspace. In calculating the covariances in the datasets, the mean image is first obtained as shown in Figure 3.8 and this is used for individual image centering for the purpose of obtaining the covariance matrix. The mean image is then subtracted from each of the activity image as shown in Equation 3.12. The fundamental approach is to compute the covariance data matrix and the eigenvectors associated with the highest eigenvalues. These top eigenvectors account for a significant representation of the entire data in the new subspace that has been formed by this transformation. With the training set of column vector X_{ij} and considering M of such vector X_{ij} ($i=1,2,\dots,M$) of length N form our training image X to ensure that a high variance is obtained with the first principal component and it is necessary that the matrix of X be properly centered. The average mean of the dataset is given in 3.11 and the centered images in 3.12 respectively.

$$\bar{X}_i = \frac{1}{q_i} \sum_{j=1}^{q_i} X_{ij} \quad (3.11)$$

$$\Phi_i = (X_{ij} - \bar{X}_i) \quad (3.12)$$

The vectors Φ_i from (3.12) are then arranged such that a new vector matrix is formed where $A = (\Phi_1, \Phi_2, \dots, \Phi_M)$. The covariance matrix of the training activity



Figure 3.8: Image Mean - Average Image Mean of All Activities

image vectors and the significant principal components of the covariance matrix can be calculated using (5) and (6) respectively.

$$C = \frac{1}{P} \sum_{i=1}^P (\Phi_i \Phi_i^T) \quad (3.13)$$

$$A^T C A = \lambda \quad (3.14)$$

where A is the matrix of the orthonormal eigenvectors of the covariance matrix C and λ represents the diagonal matrix which is the eigenvalues. Sorting the eigenvectors associated with the highest eigenvalues form a matrix A such that the newly formed matrix (feature vectors) are used in the transformation of the original datasets from one image space to a much lower dimensional new image space. Assuming that the eigenvalues are sorted in the descending order $\lambda_1, \geq, \dots, \geq \lambda_M$, the first few highest eigenvalues are then chosen. Their corresponding eigenvectors are called the principal component. The new coordinate system can be described as A where the eigenvector associated with the highest eigenvalue is chosen to be the axis of the largest variance and decreases subsequently as the eigenvalues decays. Eigenvalues that are near zero are often discarded because they are perceived to be insignificant to the recognition

3.5 Feature Extraction and Dimensionality Reduction

process. Dimension reduction is performed by keeping the eigenvector with the largest eigenvalues.

$$\Phi_i = [\Phi_1, \Phi_2 \dots \Phi_d] \quad (3.15)$$

Where d is the number of features to be determine by an application.

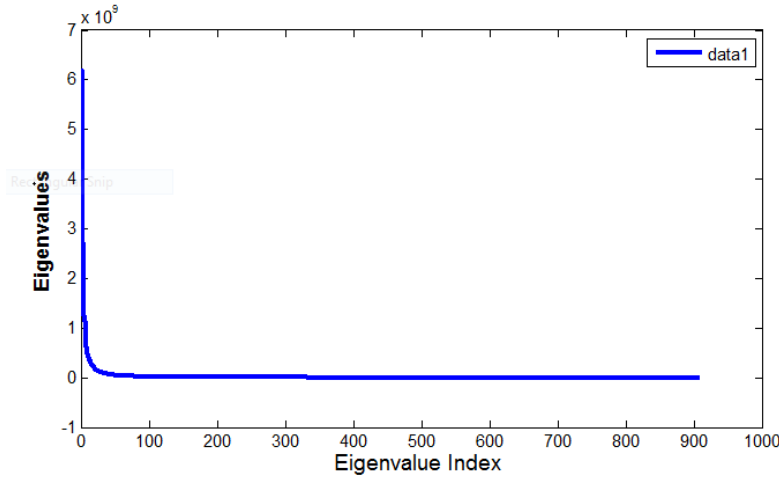


Figure 3.9: Eigenspectrum Representation - Top 910 Eigenvalues Corresponding to the Eigenvector

It can be observed from Figure 3.9 that the first few eigenvectors are the significant principal components that account for the overall variance and thus contain significant information about the data in the lower subspace. Due to the small sample set and high dimensionality in dataset, the eigenvalues as observed from the principal space of Figure 3.9 is characterized with a steeped and sharp decay. This makes the variance associated with eigenvalues in this region to be distorted, therefore become less contributive in the classification task of HAR. Similar occurrence is also expected in the complimentary and null of the eigenspectrum.

3.5.2 Fisher Linear Discriminant Analysis

Fisher Linear Discriminant Analysis (FLDA) is a supervised dimensionality reduction technique that has been used in data analysis and pattern recognition. FLDA is very useful in solving the challenges of high computational complexity and singularity problem often caused by limited training data sample. This method is known to apply the

3.5 Feature Extraction and Dimensionality Reduction

PCA method on the entire dataset first before applying the linear discriminant analysis method. This first action performed by the PCA is the reduction of the dimensionality problem common with image data. The primary purpose of LDA is to maximize the separation between different classes and minimize the separation within the same class simultaneously. The within- class and between-class scatter matrices are computed as seen in Equations 3.16 and 3.17 respectively. The between class scatter matrix S^w is defined by,

$$S^W = \sum_{i=1}^e \frac{c_i}{d_i} \sum_{j=1}^{d_i} (X_{ij} - \bar{X}_i)(X_{ij} - \bar{X}_i)^T \quad (3.16)$$

where $\bar{X}_i = \frac{1}{d_i} \sum_{j=1}^{d_i} X_{ij}$

$$S^B = \sum_{i=1}^e c_i (X_i - \bar{X})(X_i - \bar{X})^T \quad (3.17)$$

where $\bar{X} = \sum_{i=1}^e \bar{X}_i$

Forming a training set of column image vector X_{ij} , the pixel element of activity image j of person i , let the training set contains p persons and q_i be sample image for person i . For human activity recognition, each person activity is a class with prior probability c_i . The optimal discrimination projection matrix O_{LDA} that maximizes the ratio of the determinant of S^B to the determinant of S^W can be computed by solving the optimization problem.

$$O_{LDA} \operatorname{argmax} = \frac{O^T S^B O}{O^T S^W O} \quad (3.18)$$

Such that if S^W is a non-singular matrix, O_{LDA} can be maximized when the projection matrix is composed of the eigenvectors of equation (8).

$$S^{W^{-1}} S^B \quad (3.19)$$

The eigenvalue problem is solved in Equation (8) to have eigenvector matrix $O_{LDA} = A = [\Phi_1, \Phi_2, \dots, \Phi_p]$, and their corresponding eigenvalue $\lambda = \lambda_1, \lambda_2, \lambda_3, \dots, \lambda_M$. similar to that obtained in PCA. Assuming that the eigenvalues are sorted in a descending order $\lambda = \lambda_1, \lambda_2, \lambda_3, \dots, \lambda_M$, the first few highest eigenvalues corresponding to $(p - 1)$ with non-zero real eigenvalue is used for projecting the dataset for discriminant purposes.

3.6 Results obtained using PCA and FLDA on both form of motion information representation

3.5.3 Silhouettes and grayscale feature evaluation on the weizmann dataset

While we note that different models and datasets respond uniquely to feature extraction processes, we are not particularly restrained on determining on which image type is the best in general. However, this experimental setup emphasis is on the fact that both image types from the same Weizmann dataset are useful in activity recognition. Unless an exhaustive and empirical research has been conducted to prove the discriminative feature powers of each image type, this experiment cannot be misconstrued to have done such. Environmental conditions could be determining factors on the choice of image types (Grayscale, RGB or Silhouettes) and more robust hierarchical model as discussed in chapter 2, 4 and 5, are capable of learning better while encoding good feature even in harsh weather conditions. Therefore, lots of these new models are still able to function well either with any image type after necessary preprocessing. Each image set used was from the Weizmann datasets and each was partitioned into both training and testing datasets. The recognition rate outlined in this chapter is the percentage of the correct match on the testing set. PCA and FLDA models were used, while the K-Nearest Neighbour (KNN) was used as classification engine.

3.6 Results obtained using PCA and FLDA on both form of motion information representation

The result is carefully outlined in Table 3.1- 3.4. The two models were able to perform dimensionality reduction on both image types and were able to extract meaningful features for classification purposes in HAR. Firstly, while feature classification comparison was not the focus of this section, we note that the silhouettes feature on both models outperforms the grayscale features. Found in Tables 3.1 and 3.2 are the results from the PCA model and the confusion rates amongst different activities are moderately high when the grayscale images were used but such confusion had a considerable leverage in accuracy when the silhouettes features were considered. In Table 3.1, jumping and skipping had 100% accuracy. This was followed by bending, running, and waving with 90%, 86.7% and 80% accuracy respectively. The two activity that least performed in accuracy were jacking and sidewalking as they both had 66.7% and 76.7% respectively. From the results analysis, the overall performance of the PCA technique on silhouette

3.6 Results obtained using PCA and FLDA on both form of motion information representation

performed better. Compared to the PCA results shown in Table 3.2, the probable cause of this is the pixel's levels that are too similar in gray images. Similarly from Tables 3.3 and 3.4, the FLDA silhouette feature were moderately more accurate than the grayscale features set. Secondly, the FLDA model proves to be a better feature extraction model than the PCA as the accuracy of the FLDA persistently outperforms that of PCA.

Table 3.1: Confusion matrix of the recognition evaluation in % using Principal Component Analysis for silhouettes features extracted from the Weizmann database

Activities	Running	Bending	Jacking	Skipping	Sidewalk	Waving	Jumping
Running	86.7	0	0	6.7	3.3	0	3.3
Bending	0	90	0	0	10	0	3.3
Jacking	13.3	0	66.7	0	10	10	0
Skipping	0	0	0	100	0	0	0
Sidewalk	13.3	0	0	0	76.7	10	0
Waving	0	0	20	0	0	80	0
Jumping	0	0	0	0	0	0	100

Table 3.2: Confusion matrix of the recognition evaluation in % using Principal Component Analysis for grayscale features extracted from the Weizmann database

Activities	Running	Bending	Jacking	Skipping	Sidewalk	Waving	Jumping
Running	63	0	0	20	3.3	0	13.3
Bending	0	96.6	0	0	0	0	3.3
Jacking	0	0	76.6	0	13.3	10	0
Skipping	0	0	0	70	0	0	30
Sidewalk	26.6	0	6.7	6.7	53.3	0	6.7
Waving	10	0	30	0	0	36.6	0
Jumping	0	0	0	53.3	0	0	46.6

3.6 Results obtained using PCA and FLDA on both form of motion information representation

Table 3.3: Confusion matrix of the recognition evaluation in % using Fisher Linear Discriminant Analysis for silhouettes features extracted from the Weizmann database

Activities	Running	Bending	Jacking	Skiping	Sidewalk	Waving	Jumping
Running	91	2.3	0	3.3	0	3.3	0
Bending	3.3	96.3	0	0	0	0	0
Jacking	0	0	90	0	3.3	6.7	0
Skiping	6.7	0	0	93.3	0	0	0
Sidewalk	0	0	0	0	93.3	0	6.6
Waving	0	0	20	0	0	80	0
Jumping	0	0	0	0	3	0	97

Table 3.4: Confusion matrix of the recognition evaluation in % using Fisher Linear Discriminant Analysis for grayscale features extracted from the Weizmann database

Activities	Running	Bending	Jacking	Skiping	Sidewalk	Waving	Jumping
Running	90	3.3	0	3.3	0	3.3	0
Bending	0	80	0	0	20	0	0
Jacking	0	0	53.3	0	36.6	10	0
Skiping	0	0	0	100	0	0	0
Sidewalk	0	0	0	0	93.3	0	6.6
Waving	0	0	46.6	0	0	53.3	0
Jumping	13.3	6.6	0	10	0	0	70

3.7 Summary and Discussion

In this chapter 3, a clear demonstration of the importance of capturing the video information was discussed. The ability to detect such video information is key in building the entire HAR model. A simple but effective method that was used is the background subtraction method. This method has proven to be one of the reliable means of extracting video information with static background. The effect of environmental factors on the extracted video information is also highlighted and these factors are responsible for the video information degradation. A detailed process of cleaning and leveraging such disadvantageous phenomena is also proffered to ensure that a better predictive and classifying model is developed. Image enhancement methods like filtration, retaining connected components and hole filling are some of the processes used in the video cleaning. Furthermore, the video information detected is represented in this chapter as either silhouettes or the grayscale images and the silhouettes are the video information detected from the background subtraction method. The coordinates of these silhouettes were then used in cropping out their corresponding grayscale image. These two types of video information were experimented on two known methods (principal component analysis and fisher linear discriminant analysis) of feature extraction. Chapter 3 also contains an introduction to dimensionality reduction using the principal components of non-linear and uncorrelated eigenvectors of the covariance matrix of the entire dataset. The fisher linear discriminant analysis was another concept that was introduced and both methods are known for their classification and recognition strengths. Both PCA and FLDA were also used in evaluating the silhouettes and grayscale images and the results show that the silhouettes images had better recognition effects with the two models.

4

Improved Eigenspectrum Regularization for Human Activity Recognition

4.1 Introduction

With the numerous growth in machine learning and computer vision sector, there have been intense developments of different models that can improve the state-of-the-art human activity recognition scheme. Notably, chapter two extensively discussed some of these cutting-edge techniques already deployed in various fields of machine learning. Therefore, this chapter aims to further improve the work done in the area of subspace regularization to achieve better performance of human activity recognition model. This method, as diagrammatically presented in Figure 4.1, requires skillful design, correct selection of parameters and supervised classification technique to achieve better classification model. The name "hand-crafted features" is derived from this kind of model design because of direct concentration of individual design skills on each model. The holistic subspace method has gained popularity in Human Activity Recognition (HAR) in recent years, and different authors have proposed variable parameters in achieving their results [5, 7, 39].

Most subspace methods such as PCA, LDA, FLDA and other non-linear methods (KLDA and KPCA) [5–9] have been experimented in the HAR, but the problem of singularity due to small sample size has been a major issue in achieving effective feature

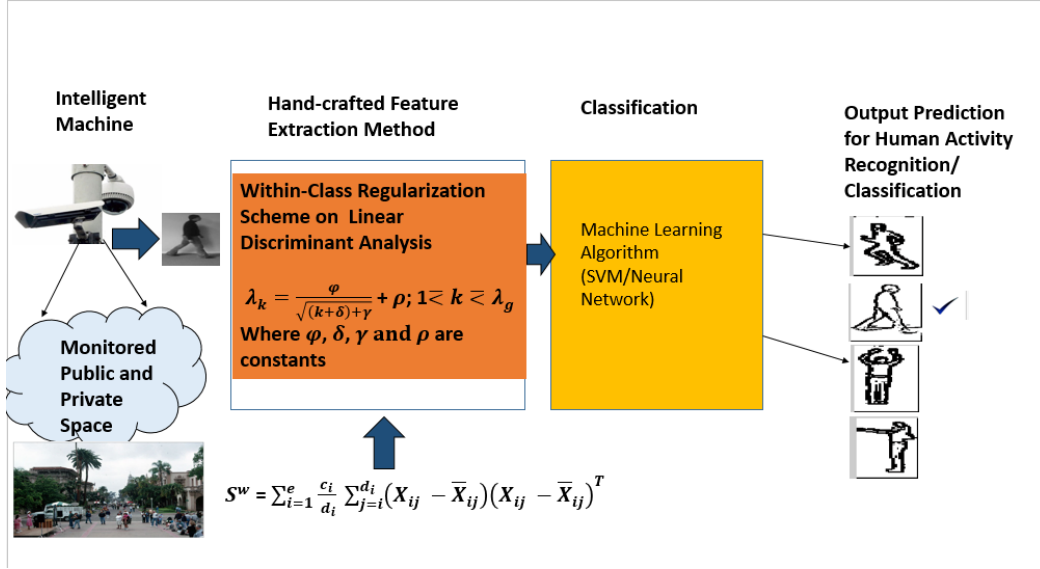


Figure 4.1: Within Class Regularization Scheme - Hand-Crafted Feature Extraction Method

extraction for the purpose of class discrimination. The decomposition of the eigenvectors in these methods are often ranked from the most significant to the least significant. The ranking of these eigenvectors helps capture and separate the most significant eigenvectors (principal components) that show strong variance and direction in the pattern of the entire data. Other eigenvectors down the list of the ranked eigenvectors are the ones with the least information regarding the variances in the data. The common practice adopted in most subspace methods of recognition is to discard those eigenvectors deemed to be insignificant in the recognition process [180]. The concept of discarding eigenvectors associated with smaller eigenvalues developed by most subspace methods have made them a poor recognition model, notably the loss of information no matter how small will certainly result in poor classification results [29]. Hence, the concept of feature regularization was developed, thus instead of discarding off small eigenvectors values considered to be insignificant, the regularization scheme attempts to optimize the eigenvectors. The regularization scheme constrain the recognition model to function much closely to what would have been achieved without the effect of the problem of the small sample size. There have been huge efforts and relentless studies by researchers to proffer better methods of feature regularization in the field of computer vision. But as discussed in the literature review chapter, there have been some shortcomings in the

study of effective feature discrimination [5, 7, 27–30, 181]. In general, the most sought-after features are those that can present the minimum within-class variations and the maximum between-class variations. However, because the within-class variances are extracted based on small sample size of the training images, it becomes very difficult to actualize the minimum variations presented by the within-class scatter matrix. The variances obtained from this kind of problematic within-class scatter are unbalanced and they also have the inclination to overfit a particular training dataset. The unpleasant complications developing from the within-class scatter is a major reason accounting for why the subspace method used for human activity recognition results in poor recognition performance [5]. The need to work and improve the within-class scatter usability to improve HAR cannot be overemphasized and this underscores the importance of its regularization. This study presents within-class subspace regularization methodology to promote effective and better feature extraction for HAR. This regularization process is an improvement on the work done by [5]. In [5] approach, the image space traversed by the eigenvectors of the within-class matrix is disintegrated into three subspaces and each subspace is regularized differently. The mode of dividing these subspaces and how to determine the start and end point of each subspace remains an open question for further studies. Determining the subspace components like principal, noise, null space and regularizing them separately is very cumbersome and prone to errors. Therefore, this study proposed the usage of more eigenvalues from the dependable subspace to achieve a four-parameter modelling system. In addition, the regularization process is done in one whole piece and by so doing circumvents the difficulties experienced in subspace decomposition. This model allows an improved and better projection of the eigenvectors that are biased by small sample size influence. This regularization is computed in one piece thereby circumventing unnecessary difficulties of modeling eigenspectrum in an otherwise segmented manner. The entire eigenspace is utilized for extracting quality features and thus prevents discarding features that can be used for discrimination. Feature extraction and dimensionality reduction is finalized at a future phase of the appraisal stage of recognition.

4.2 Eigenspectrum Modeling

In representing the various forms of human activities with a set of images, the training set formed consists of a column vector $\{X_{ij}\}$, where $X_{ij} \in R^{(n=hw)}$ and R is defined to be the image vector. The 2D training dataset from the human activities are vectorized such that they become an 1D vector concatenated column wise [67]. Consequently, the entire training image is reassembled into a vector of length $R = hw$, and it is worth nothing that the M number of such vector $\{X_{ij}(i = 1, 2, \dots, M)\}$ comprises e events and d_i be number of images in the i events, and c_i being considered as the prior likelihood. Equation 4.1 denotes the entire number of training samples in the datasets.

$$T^n = \sum_{i=1}^e d_i \quad (4.1)$$

$$S^W = \sum_{i=1}^e \frac{c_i}{d_i} \sum_{j=1}^d (X_{ij} - \bar{X}_i)(X_{ij} - \bar{X}_i)^T \quad (4.2)$$

where $\bar{X}_i = \frac{1}{d_i} \sum_{j=1}^{d_i} X_{ij}$

$$S^B = \sum_{i=1}^e c_i (X_i - \bar{X}_i)(X_i - \bar{X}_i)^T \quad (4.3)$$

while 4.2, 4.3 and 4.4 represent the within-class, between class and total class matrix respectively:

$$S^{Total} = \sum_{i=1}^e \frac{c_i}{d_i} \sum_{j=1}^d (X_{ij} - \bar{X})(X_{ij} - \bar{X})^T \quad (4.4)$$

where $\bar{X} = \sum_{i=1}^e \bar{X}_i$ Let δ represents one of the above scatter matrix, then we can solve the eigenvalue problem as shown in equation 4.5

$$A^T \delta A = \lambda \quad (4.5)$$

In solving the eigenvalue problem, a decomposition of the image vector and class mean vector constitutes a linear transformation of the image vector. This linear transformation produces eigenvectors and their corresponding eigenvalues. The eigenvectors associated with the scatter matrix δ are denoted as $A = [A_1, A_2, \dots, A_M]$, and its corresponding eigenvalues λ which are seen in the matrix diagonal are represented as $\lambda = \lambda_1, \dots, \lambda_M$. Organizing these eigenvalues in a descending order $\lambda_1, \geq, \dots, \geq \lambda_M$ and

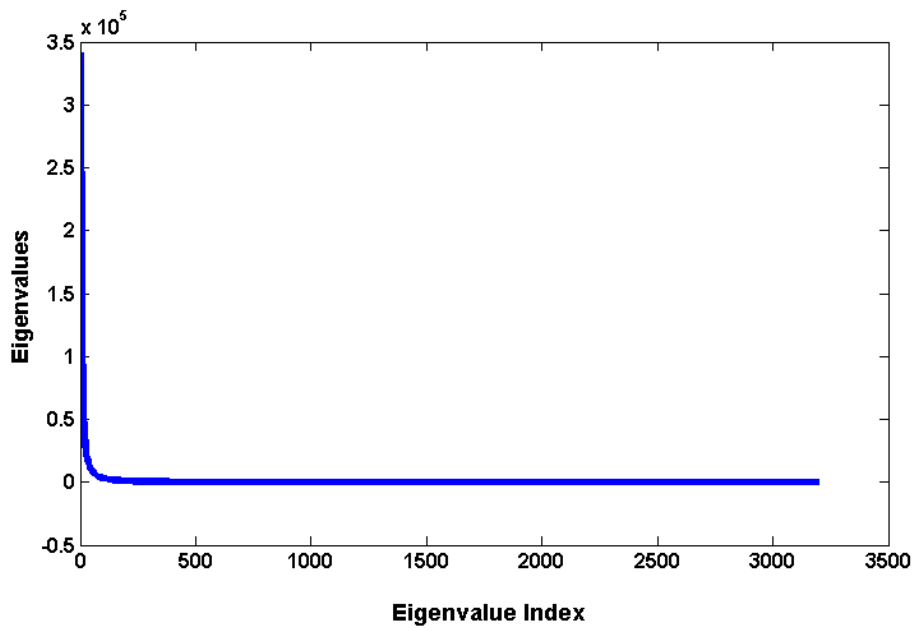


Figure 4.2: Eigenspectrum obtained from the training data - A Plot of Eigenvalues Against Eigenvalue Index

plotting the same against its index K results in the graph called the eigenspectrum of a class matrix as can be seen in Figure 4.2. The extraction and scaling of the features are vital parts in subspace method as this technique allows for easy representation of the features in a different space and provides better discriminative capacity amongst dissimilar class. Thus, λ is a notable tool used to scale and extract features and thus underscores its importance in subspace model [5].

4.2.1 The Problem of unregularised subspace

The use of subspace technique for the recognition of HAR will be sub-optimal in its performances even with accurate representation of training set data. The underachievement is primarily due to the presence of both the within-class and between class matrix of the image data. However, the within-class matrix has been identified as the crucial entity that poses many complexities and challenges in the subspace method for recognition [7, 184]. These challenges result from the singularity problem which is a phenomenon where the dataset have very high dimensionality but small sample size. The presence of small and near zero eigenvalues in the eigenspectrum plot has been identified as one of the main drawbacks presented by the singularity problem [27, 182–184].

With succinct literature reviews, efforts made on regularization were concentrated on the within-class matrix, and it was discovered that the within-class matrix is responsible for the poor recognition performance[5, 67]. The swift decay of the eigenspectrum in Figure 4.2 is due to the problem of the small sample size, occurrence of noise in the datasets, high dimensionality and correlation in image pixels. The within-class matrix in sympathy with these constrains suffers from under representaion and invariably poor performance. The challenges presented by the within-class matrix to recognition model is enormous. The introduction of very small and zero eigenvalues in the eigenspectrum plot is a result of the small sample size phenomenon. Hence, it is difficult for a model with unregularized within-class scatter to achieve effective discriminative power and as such this presents poor discriminative capability in the training set of different classes, while it fails to exhibit lower variance with similar class activity. Unregularized subspace presents two peculiar problems in the computer vision task. Firstly, scaling eigenvectors with very small or zero eigenvalues promotes eigenvector to be appropriated with undue weight and this may lead to overfitting and poor model generalization.

This presents deviations and unexplained accuracy of results in the recognition process. Secondly, harnessing principal eigenvalues and neglecting smaller eigenvalues is comparable to the weighting of features with a step function activation function as seen in 4.6. This hypothesis presents this technique with a forfeiture of important discriminatory features [29].

$$t_k^m = \begin{cases} 1, & k \leq M \\ 0, & k > M \end{cases} \quad (4.6)$$

where t denotes M principal eigenvalues.

In [5], the benefits of controlled eigenvectors scaling was discussed and elaborated. The λ_k is valuable and significant in feature scaling and extraction, the parameter $\sigma_k = \sqrt{\lambda_k}$, which represents scaled eigenvalues is invaluable to buiding the HAR model. Eigenvalues weighting is done by scaling the eigenvalues by a factor $\frac{1}{(\sqrt{\lambda_k})^{0.7142}}$ as shown in 4.7, while the whitening procedure is done by normalizing the eigenvector by the weighting function as seen in 4.8.

$$W^k = \frac{1}{(\sqrt{\lambda_k})^{0.7142}} \quad (4.7)$$

where $k = 1, 2, \dots, n$.

$$A^s = [A_k w_k, \dots, A_n w_n] \quad (4.8)$$

A^s in equation 4.8 is the scaled eigenvectors used in transforming the dataset into a different feature space.

4.2.2 Effect of Using Small Eigenvalues to Scaled Eigenvectors

The use of small eigenvalues for eigenvectors whitening certainly have unique consequences of projecting undue and badly scaled eigenvector. Since the eigenvalues shrink in values as it spreads down the eigenspectrum graph as shown in Figure 4.2, attempting to scale small and zero eigenvalues with exponential inverse function as shown in 4.7 leads to high noise level. Again, badly scaled eigenvectors that can cause overfitting is another critical imbalancing introduced to the model. The use of such weighting function presents an unjustifiable scaling of small eigenvalues along the eigenspectrum

bottom; hence unregularized eigenvalues are susceptible to errors and poor recognition model. To substantiate the narrative of this unwholesome scaling effect of small and zero eigenvalues, a graphical illustration is shown in Figure 4.3. Figure 4.3 shows a steady rise along the eigenvalue index and this steady rise at one point suddenly plunged down to zero. This illustrates a clear example of how small or zero eigenvalues can distort and disrupt useful eigenfeatures of the within-class matrix (S^W) that are highly needed for HAR purposes. Hence, poorly scaled models such as this results in misclassifications that are undesirable. Equation 4.9 further illustrates the step function characteristics of unregularized eigenvalues.

$$W_k^M = \begin{cases} \frac{1}{(\sqrt{\lambda_k})^{0.7142}}, & k \leq r_m \\ 0, & r_m < k \leq M \end{cases} \quad (4.9)$$

Where r_m is the ranks of the within-class scatter matrix and $M \geq r_m$.

Unregularized scaling of these eigenvalues summarizes the eigenvalues in the complementary null space to be zero, though it contains important discriminative features that are useful for classification. Therefore, the studies seeks to reduce or eliminate the negative effect of unreliable and small eigenvalues that are common in subspace method by regularization technique. This use of the regularization technique is necessary, this is because eigenvalues and their corresponding eigenvectors of the within-class matrix does not correctly reflect the true variance of the within-class matrix. [5].

4.2.3 Extrapolation and Modeling of Within-Class Matrix Using Four-Parameters

The presence of noise down the eigenspectrum is not desirous because of its negative effect of distorting features true estimation that is needed to build HAR model. Adequate extraction and representation of features is vital to successfully build HAR recognition systems. Hence, it is important to invest on the feature engineering process that includes featutre modelling. The principal eigenvalues and their corresponding eigenvectors of the within-class matrix are the most informative and significant in describing components of human activities [5, 6, 185, 186]. As seen in Figure 4.2 of the eigenspectrum, these first few eigenvalues which constitute the start of the principal segments are better reflection of the true variances exhibited in the image matrix. They

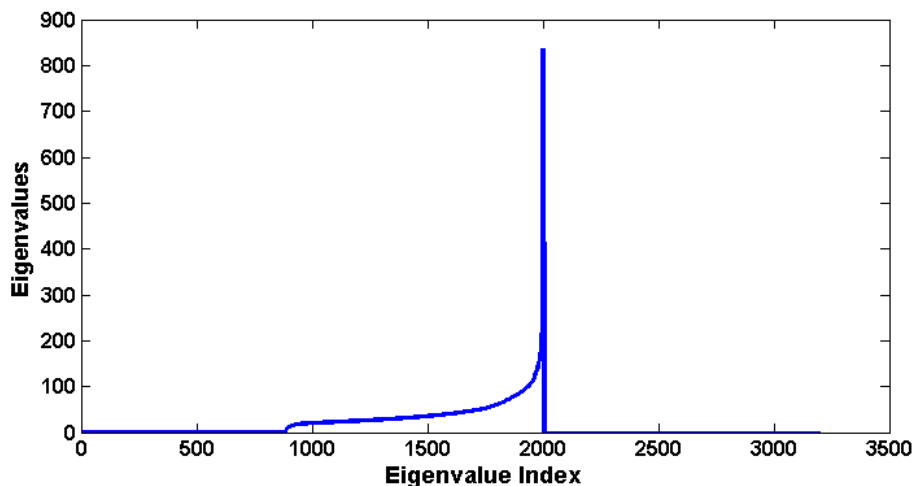


Figure 4.3: Unregularized Eigenspectrum - Weighted Eigenspectrum obtained from unregularized training data

capture and encode important information about the data presented. These generative techniques help in prediction and modeling of the eigenvalues that had lost values down the eigenspectrum because of the problem of singularity. The exact and robust modeling scheme of these first few eigenvalues ensure accuracy and succinct prediction of eigenvalues down the eigenspectrum. This estimate enables small and zero eigenvalues that constitute the source of unreliability to be substituted by more reliable and dependable predicted values. For the purpose of predicting reliable eigenvalues down the eigenspectrum, a four parameter constant modeling scheme is developed such that the shortcomings of unregularized eigenvalues is addressed. These parameter constants are $\varphi, \sigma, \gamma,$ and ρ bounded by the constraint $\lambda_k \mid 1 \leq k \leq \lambda_g$. The emphasis is on the approximation of the remaining portion of the eigenvalues after a successful modeling of the reliable portion at the start of the eigenspectrum has been determined. In these studies, an attempt to capture the true variances from the reliable space using a four-parameter constant is the main goal. λ_g is taken to be the upper bound of the top (1%) of the eigenvalue length (k). Based on this hypothesis, approximating and deducing the outstanding ninety-nine (99%) of the eigenspectrum can be predicted from the modeled first few eigenvalues with the help of our four parameter constants. The first few k from the eigenspectrum is thus modeled by 4.10

$$\lambda_k = \frac{\varphi}{\sqrt{(k + \sigma) + \gamma}} + \rho, \quad 1 \leq k \leq \lambda_g \quad (4.10)$$

where $\varphi, \sigma, \gamma,$ and ρ are constants. The formulation of these parameters is derived to mirror the characteristics related with the variances in the first few eigenvalues. Mandal and How-Lung [5, 187] investigated the effect of using three parameters. Their method tailored along the scope of having all the eigenvalues spectrum partition into three different subspaces and then regularized differently. The complexity of trying to determine the start and end of each subspace can be very frustrating and difficult to achieve especially for a large data set. Therefore, the present model proposed a non-piecewise regularization approach, thereby avoiding the complexity associated with partitioning of the subspace into different sections. All extrapolated eigenvalues are used to replace distorted eigenspectrum. The new extrapolated eigenvalues are for regularization and extraction of valuable eigenfeatures that would have been discarded or difficult to extract. The parameter $\varphi, \sigma, \gamma,$ and ρ in equations 4.11 - 4.14 are derived by $\lambda_D = \lambda_{k=1}$, $\lambda_g = \lambda_{1\%k}$, $g_1 = \frac{\sqrt{g}}{2}$ and g is the index position for the λ_g value

$$\gamma = \frac{(\lambda_D - \lambda_g) - (\frac{\lambda_g}{2})\lambda_g - (\frac{\lambda_g}{2} - \lambda_g) - \lambda_D}{(\frac{\lambda_g}{2})\lambda_g + \sqrt{(\lambda_D \frac{\lambda_g}{2}) + (\lambda_D \frac{\lambda_g}{2})}} \quad (4.11)$$

$$\sigma = \frac{\sqrt{\lambda_D \lambda_g}}{\frac{\lambda_g}{2}} \quad (4.12)$$

$$\varphi = \frac{(\lambda_D - \lambda_g)(1 - \gamma)(\sqrt{g} + \gamma)}{\sqrt{g} - 1} \quad (4.13)$$

$$\rho = \lambda_D - \frac{\varphi}{1 + \gamma} + \frac{1}{g} \quad (4.14)$$

The fast and swift decay experienced by the eigenspectrum as shown in Figure 4.2 illustrates a flawed eigenspectrum, a notable reason for poor recognition performance. In Mandal and How-Lung [5, 187], an attempt was made to proffer solution to this problem. Although results recorded some improvements, there was still room for further exploration. With the introduction of a fourth parameter, it is clear from Figure 4.4 that dependable and more precise eigenvalues have been reproduced. Again, a closer look at Figure 4.5 shows that a substantial improvement at lowering the eigenspectrum rapid decay has been achieved relative to both the unregularized and three parameter modeling curves respectively.

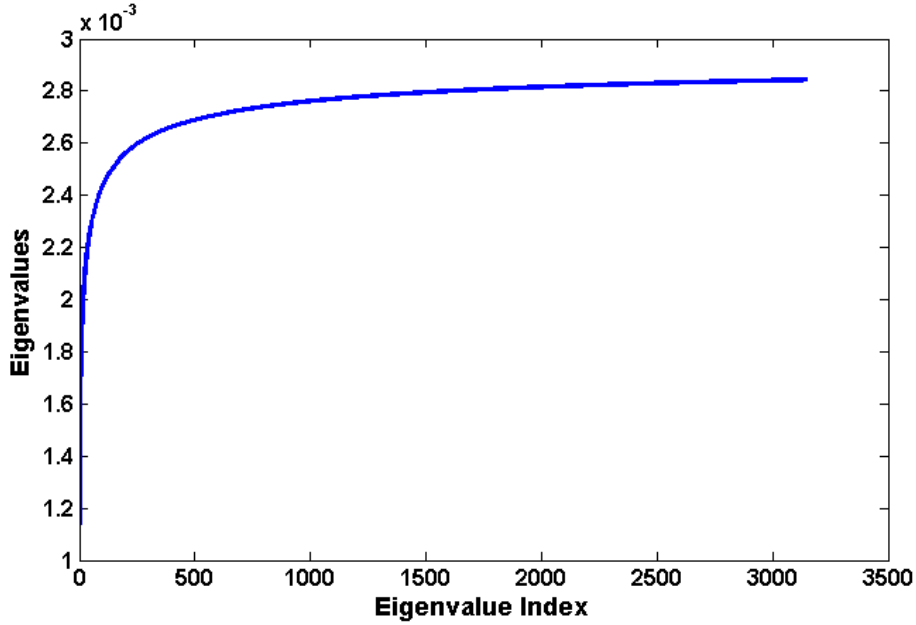


Figure 4.4: Regularized Eigenspectrum - Regularized Weighted Eigenspectrum obtained from training data

Figure 4.5, shows the eigenspectrum of unregularized, regularized 3-parameter and 4-parameter eigenvalues. The analysis of Figure 4.5 graph reveals that the unregularized eigenspectrum has a sharp decay curve, while that of the 3-parameter modeling scheme has a reduced curve. A further improvement in the decay curve was observed with the use of 4-parameter constants. The quick and rapid decay has been reduced drastically with the introduction of the 4-parameter modeling techniques. The results obtained demonstrate that this regularization scheme provides a better and stable use of the subspace method with the within-class matrix.

4.3 Regularization and Extraction of features

Equation 4.10 is for the eigenvalues modeling and the modeled eigenvalues are used for the eigenvector regularization process. Since most of the variances associated with the datasets are found in the principal component of the within-class matrix, the extrapolation of the real eigenvalues is determined from the modeled eigenspectrum. Therefore, an efficient regularization of within-class scatter matrix was achieved in this study. A variable, if unregularized, presents noisy eigenvalues responsible for poor classification

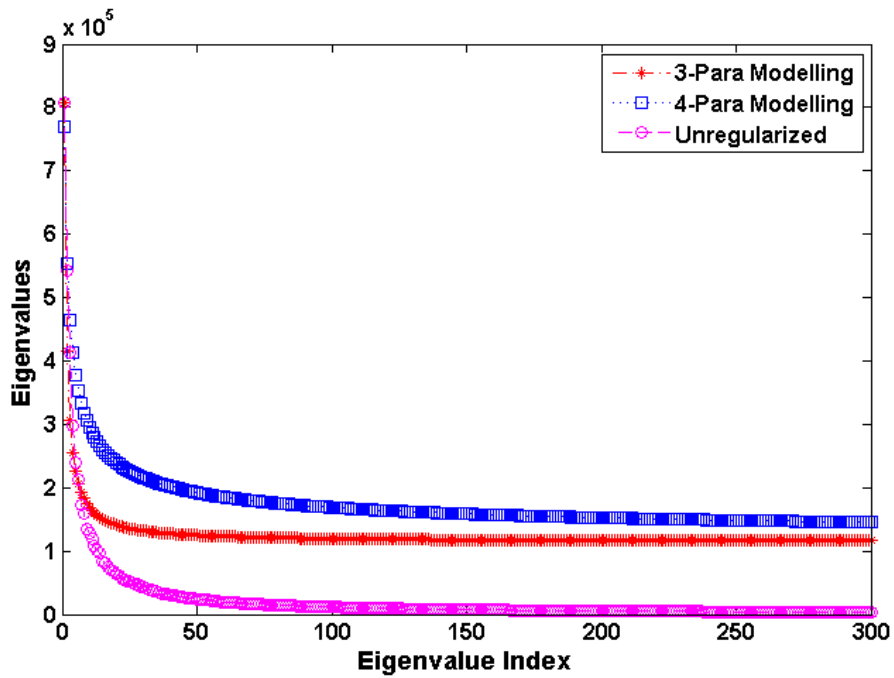


Figure 4.5: Different Eigenspectrum Modelling - Eigenspectrum of unregularized, 3-parameter and 4-parameter modelled eigenvalues

4.3 Regularization and Extraction of features

and recognition when subspace method is used. The estimated whitened eigenvalues as seen in Equation 4.8 are used to scale the eigenvectors in order to capture and promote its discriminative ability. Such eigenvalues can be used for eigenfeature scaling without adversely affecting the recognition model built for HAR. Equation 4.15 is used for the feature scaling and extraction while Equation 4.16 is the modeled eigenspectrum. The complications and distortions that can arise from unregularized eigenfeature underscores the need to address the problem that small and near zero eigenvalues present. Figure 4.3 interprets and demonstrates the steps of the function character and operates by converting eigenvalues less than a threshold minimum to zero thereby discarding useful features needed for discrimination. The regularization scheme has helped proffer a solution to the menace of the small and zero eigenvalues caused by noise and small sample size. The undesirable problem of the step function in Figure 4.3 has been addressed with the 4-parameter eigenspectrum modelling. This is seen by a continuous steady rise in Figure 4.4. In Figure 4.5, the 4-parameter regularization of the within-class matrix enabled stable and steady decay of the eigenspectrum. This graph and other emperical findings demonstrate that the 4-parameter regularization technique is better than unregularized and 3-parameter constant method of regularization.

$$A^s = \phi k W_k, \dots \phi n W_n] \quad (4.15)$$

$$\lambda_k = \frac{\varphi}{\sqrt{(k + \sigma) + \gamma}} + \rho, \quad 1 \leq k \leq \lambda_n \quad (4.16)$$

where, $k = 1, 2, 3 \dots n$ and A^s is the scaled eigenvectors for transforming the training data into another feature space.

$$U_{ij} = (A^s)^T X_{ij} \quad (4.17)$$

The transformation of the entire data is done by the scaled eigenvectors obtained from the within-class matrix given in Equation 4.15, while the new space U_{ij} is represented in Equation 4.17. The main advantage of this method is that the transformed data U_{ij} dimension is the same as the training data. With this development, the issue of dimensionality reduction does not arise and hence all the extracted eigenvalues are useful in the eigenfeature extraction purpose. The new data we have is U_{ij} , and considering the formation of total scatter matrix from the new data space, this provides

4.4 Experimental Result and Discussion

the model with a maximum discriminant feature. The total scatter matrix developed from the new data U_{ij} is given in Equation 4.18.

$$S^{Total} = \sum_{i=1}^e \frac{c_i}{d_i} \sum_{j=1}^d (U_{ij} - \bar{U})(U_{ij} - \bar{U})^T \quad (4.18)$$

where $\bar{U} = \sum_{i=1}^e \frac{c_i}{d_i} \sum_{j=1}^d U_{ij}$. If we solve the eigenvalue problem in 4.18 and retains the most significant features by keeping eigenvectors ϕ_Z with the z largest eigenvalues,

$$\phi = [\phi_k]_{k=1}^z \quad (4.19)$$

The dimension of the feature space is only reduced at this point, and more feature extraction is possible using equation 4.20.

$$Y = A^s \phi_z \quad (4.20)$$

From equation 4.20, Y is the new feature vector used to transform the training image into a more discriminative space as seen in 4.21.

$$Q = Y^T X_{ij} \quad (4.21)$$

To validate our theoretical and empirical analysis, the K- Nearest Neighbour (KNN) distance measure and Artificial Neural Network (ANN) were used to validate the effectiveness of the proposed 4-parameter eigenfeature regularization and extraction method.

4.4 Experimental Result and Discussion

The database used in this study are the Weizmann and KTH datasets, the Weizmann dataset contains 10 classes of activities, however seven of these action classes were considered for this model evaluation. The seven actions involved are running (run), bending (bend), jumping jack (jack), skipping (skip), galloping sideways (side), waiving-two hands (wave) and jumping forward-on-two-legs (jump). A total of nine persons was involved in the performance of these seven activities and five of the nine characters were used for training purposes while the remaining four were used for testing. The training set obtained from the Weizmann datasets consist of 1450 images and 380 testing images and the experimental processed images were resized to 80×40 pixels. From the experimental results analysis as shown in Table 4.1 -4.4, an outstanding

4.4 Experimental Result and Discussion

recognition accomplishment demonstrated by the new 4- parameter regularization and extraction technique are shown. Our proposed technique has demonstrated state-of-the-art superiority over other subspace methods like PCA, FLDA, and the recently state-of-the-art 3-parameter eigenfeature regularization method. Benchmarking the present method with the recent state-of-the-art [5], has strengthened the superiority of this 4-parameter eigenfeature regularization and extraction method and this is evident in succinct comparison between Tables 4.3 and 4.4.

The poor classification demonstrated by the PCA, FLDA (Tables 4.1 - 4.2) is partly because of the earlier explained step function characteristics that are mostly common with unregularized eigenvalues. Again, the small sample size problem which is common to most dataset is also a reason for poor generalization. Therefore, this proposed model has added innovation to the 3-paramater structure earlier proposed by improving the method of eigenfeature regularization to aid the extraction of quality features that are important in building the HAR model.

The procedure by which a median outlier method was used by the 3-parameter model to determine the beginning of the reliable and unreliable subspace is very cumbersome and prone to errors in feature extraction and regularization process. Hence, the motivation behind our method of extrapolating the remaining eigenvalues at one piece. This is possible when the reliable modeling of the principal eigenvalues has been established. This process stops unnecessary disintegration of the eigenspectrum into different subspaces before regularization. Table 4.4 shows better recognition than other subspace methods discussed in chapter 2. To further buttress the recognition strength derived from the 4-parameter eigenvalue regularization and extraction model, the artificial neural network (ANN) classifier is used for the training and classification of these features. The results as shown in Tables 4.5 - 4.6 validate the developed 4-parameter model's efficacy than the earlier known methods and that of the recent 3-parameter feature regularization and extraction. With the neural network classifier, the overall performance rate of 97.1% was achieved with the proposed technique compared to 94.3% achieved by using the 3-parameter for feature regularization and extraction method.

4.4 Experimental Result and Discussion

Table 4.1: Confusion matrix of the recognition evaluation in % using Principal Component Analysis for different activities of the Weizmann database

Activities	Running	Bending	Jacking	Skiping	Sidewalk	Waving	Jumping
Running	63	0	0	20	3.3	0	13.3
Bending	0	96.6	0	0	0	0	3.3
Jacking	0	0	76.6	0	13.3	10	0
Skiping	0	0	0	70	0	0	30
Sidewalk	26.6	0	6.7	6.7	53.3	0	6.7
Waving	10	0	30	0	0	36.6	0
Jumping	0	0	0	53.3	0	0	46.6

Table 4.2: Confusion matrix of the recognition evaluation in % using Linear Discriminant Analysis for different activities of the Weizmann database

Activities	Running	Bending	Jacking	Skiping	Sidewalk	Waving	Jumping
Running	90	3.3	0	3.3	0	3.3	0
Bending	0	80	0	0	20	0	0
Jacking	0	0	53.3	0	36.6	10	0
Skiping	0	0	0	100	0	0	0
Sidewalk	0	0	0	0	93.3	0	6.6
Waving	0	0	46.6	0	0	53.3	0
Jumping	13.3	6.6	0	10	0	0	70

Table 4.3: Confusion matrix of the recognition evaluation in % using three parameter modeling for different activities of the Weizmann database

Activities	Running	Bending	Jacking	Skiping	Sidewalk	Waving	Jumping
Running	60	3.3	3.3	3.3	6.7	6.7	16.7
Bending	0	87	0	0	3.3	10	0
Jacking	0	0	76.6	0	6.7	13.3	3.3
Skiping	3.3	3.3	6.7	83.3	0	3.3	0
Sidewalk	13.3	0	3.3	3.3	66.7	13.3	0
Waving	0	0	20	0	0	80	0
Jumping	6.7	0	0	3.3	0	0	90

4.4 Experimental Result and Discussion

Table 4.4: Confusion matrix of the recognition evaluation in % using four-parameter eigenfeature regularization and extraction from various activities of the Weizmann database

Activities	Running	Bending	Jacking	Skipping	Sidewalk	Waving	Jumping
Running	80.3	3.3	6.7	0	6.7	0	3
Bending	0	96.6	0	0	3.3	0	0
Jacking	0	0	94	0	0	6	0
Skipping	10	0	0	90	0	0	0
Sidewalk	6	0	0	0	90	4	0
Waving	0	0	0	0	0	100	0
Jumping	13.3	0	0	0	0	0	86.7

Table 4.5: Confusion matrix of the recognition evaluation with our three parameter using the ANN classifier on the Weizmann database

Activities	Running	Bending	Jacking	Skipping	Sidewalk	Waving	Jumping	Percentage
Running	20	2	0	2	0	0	0	83.3
Bending	0	20	0	1	0	0	0	95.2
Jacking	0	0	19	0	0	0	0	100
Skipping	0	0	0	16	0	0	0	100
Sidewalk	0	1	0	0	19	0	0	95.2
Waving	0	0	0	0	0	20	0	100
Jumping	0	0	1	0	0	0	16	95.2
percentage	100	90.9	95.2	88.9	100	100	100	94.3

4.4.1 Results on the KTH Database

With the KTH database, Images from the video frames are extracted and preprocessing is applied to the extracted image sequence. The KTH datasets comprise six different kinds of human actions and these actions are: walking, jogging, running, boxing, hand waving, and hand clapping. These grayscale images are then cropped to 90×35 pixels. A total of 1200 images were used for the training and 210 images for testing. The confusion matrix in Tables 4.7 - 4.10 highlight the recognition level of each model discussed in this thesis. The proposed 4-parameter extraction and regularization method have shown great recognition result than those of PCA, FLDA and the most recent state-of-the-art 3-parameter extraction and regularization method. The advantage of regularizing and harnessing the lower ranked eigenvectors that were hitherto an obstacle in

4.4 Experimental Result and Discussion

Table 4.6: Confusion matrix of the recognition evaluation with our four parameter using the ANN classifier on the Weizmann database

Activities	Running	Bending	Jacking	Skiping	Sidewalk	Waving	Jumping	Percentage
Running	20	2	0	2	0	0	0	83.3
Bending	0	20	0	0	0	0	0	100
Jacking	0	0	20	0	0	0	0	100
Skiping	0	0	0	16	0	0	0	100
Sidewalk	0	0	0	0	21	0	0	100
Waving	0	0	0	0	0	20	0	100
Jumping	0	0	0	0	0	0	16	100
percentage	100	90.9	100	88.9	100	100	100	97.1

other subspace methods clearly demonstrates the indispensability of this 4 parameter technique. Again, a comparison of the Tables 4.9 and 4.10, even though the principle of regularization that existed between these two is somewhat similar but different, the superiority of the 4-parameter methods over weight that of the 3-parameter modeling methods. The discarding of lower ranked eigenfeatures as practiced by other subspace methods was completely disregarded in this method. The observation seen in Table 4.10 is that jogging and running have slight misclassification. This is because of the spatiotemporal correlation that exists between the three class activities of walking, running, and jogging. Similar observation is also seen in boxing, hand-waving, and hand clapping.

Table 4.7: Confusion matrix of the recognition evaluation in % using Principal Component Analysis for different activities of the KTH database

Activities	Walking	Jogging	Running	Boxing	Hand Waving	Hand Clapping
Walking	68	8	3	0	7.5	20
Jogging	33	18	10	0	15	24
Running	2.5	10	60	0	20	7.5
Boxing	7	0	18	30	27.5	17.5
Hand Waving	0	0	0	0	40	60
Hand Clapping	10	0	8	0	27	55

4.4 Experimental Result and Discussion

Table 4.8: Confusion matrix of the recognition evaluation in % using Linear Discriminant Analysis for different activities of the KTH database

Activities	Walking	Jogging	Running	Boxing	Hand Waving	Hand Clapping
Walking	67	8	15	1	1	8
Jogging	0	72	8	8	0	12
Running	7.5	6	68	0	7.5	11
Boxing	15	5	0	70	7	0
Hand Waving	0	10	10	0	60	20
Hand Clapping	17	3	11	0	16	63

Table 4.9: Confusion matrix of the recognition evaluation in % using three-parameter modeling for different activities of the KTH database

Activities	Walking	Jogging	Running	Boxing	Hand Waving	Hand Clapping
Walking	100	0	0	0	0	0
Jogging	4	88	8	0	0	0
Running	0	24	76	0	0	0
Boxing	12	4	8	76	0	0
Hand Waving	16	0	0	0	84	0
Hand Clapping	0	0	0	0	0	100

Table 4.10: Confusion matrix of the recognition evaluation in % using four-parameter eigenfeature regularization and extraction from various activities of the KTH database

Activities	Walking	Jogging	Running	Boxing	Hand Waving	Hand Clapping
Walking	100	0	0	0	0	0
Jogging	0	94	3	3	0	0
Running	0	3	94	3	0	0
Boxing	0	3	0	94	2.8	0
Hand Waving	0	0	3	0	97	0
Hand Clapping	0	0	0	0	0	100

4.5 Summary and Discussion

The new 4-parameter regularization method has shown promising insight in the new frontiers of computer vision sector especially in the recognition and classification of human activities. This proposed technique has the advantage of becoming a competent extraction means of highly discriminative features by applying a regularization process. The regularization process is unique which is a one-piece regularization of the entire eigenspectrum in order to avoid the difficulties posed by isolated fragmentation of the eigenspectrum subspace before regularization has been overcome. Therefore, this research work has not only proposed a better model with the 4-parameter, but has also given important insight on how the use of within-class matrix in subspace recognition can drastically affect the classification model in computer vision tasks. The 4-parameter has enabled the creation of a more reliable, dependable and accurate eigenvalues for prediction purposes. With these regularization techniques, the within-class matrix is seen as an important term in the recognition of human activities. The discussion of the results shown on the Weizmann and KTH databases are substantiated by asserting that the 4-parameters regularization method is a key connection in the mining of discriminant information. Considering all our experiments and results, the proposed 4-parameter approach has shown better discriminative power amongst activities that are similar than other popular discriminative methods such as the PCA, FLDA which is inclusive of the new state-of-art 3 parameters regularization and extraction method. The future direction of this research is the implementation of this cutting-edge method to the fastest growing 3-D data images. Current active research in 3-D imaging has stood out to revolutionize the perception of a real-world implementation of autonomous vehicles, augmented reality and other fast-growing computer vision tasks that need superior accuracy in their inference machine.

5

Deep Learning and Feature regularization in Convolutional Architecture for Human Activity Recognition

5.1 Introduction

The concept of building shallow networks and creating handcrafted features for the recognition of human activity is receiving less attraction because of a change in paradigm of building deep learning models for purposes of recognition. Convolutional neural networks (CNNs) which comprises one of the deep learning methods is ubiquitous in the area of machine learning and artificial intelligence. This method has been widely celebrated for its unique power to extract discriminative features for recognition and classification purposes. Unlike the shallow architectures with a simple layer capable of performing non-linear feature transformation, deep layer architecture is more complex. Notably, deep layer architectures are known for learning convoluted non-linear and salient features that cannot be learned by simple network or handcrafted features. Again, another common advantage of this architecture is its ability of learning in an end-to-end manner; thus, providing a means of high-level representation of hidden patterns in the datasets. As the field of computer vision evolves, its numerous applications in almost every area of our lives have made it important to develop credible algorithm

to push further the boundaries of machine vision. This development is a conscious attempt to make machines have better perception of their environment and make necessary inference in the same way as humans do. Deep learning models are built to mimic the functionality of the layers of the neurons found in the neocortex which is a part of the brain where higher-order function such as sensory perception and all other cognitive processes are coordinated. Therefore, deep learning is a landscape of artificial neural networks capable of learning data representation. While there are different variants of deep learning models as discussed in the literature section, the choice of any deep learning variant depends on the functionality of the parameter and areas of application. Thus, this chapter comprehensively explore the CNNs, improvement of their discrimination processes of the features, features extraction method and their deliverables in building models that can efficiently recognize human activities. While we acknowledge the success attained in extracting good features with the use of the deep learning method, feature learning, optimization, parameter and hyperparameter selection are vital for improving the deep learning model performance. Overfitting models are common problems that are also associated with deep learning methods like CNNs. Feature regularization is an important step to ensure that any deep learning model does not only generalize well to unseen data but also maintains high levels of accuracy in the recognition and classification task. With CNNs, the softmax loss is used as the traditional loss function. This loss function allows deep features of distinct classes to be separated and promotes effective training of deep neural network. An improvement on CNNs discriminative power for facial recognition was recently reported where softmax and center loss are jointly used as supervisory loss signal. It is shown that such supervisory loss function is not optimum in human activity recognition. Hence, a new likelihood regularization term is used to improve the feature discriminative power of the CNNs model. A new regularization term that can improve class discrimination is introduced. This regularization term is modeled from Bayesian distribution priori for posterior estimation of class probability density. A quick overview of other forms of regularization, like dropout and L2 regularization, will be highlighted and integrated with the developed model. The summary effect of the dropout on neural networks is a method which light-loosens the composition of co-adapted features. This process will disable other retained neuron components from contributing their overall weighting

5.2 Architectural Design of Convolutional Neural Network

process. The effect of dropping incoming and outgoing weight connection, which influences model generalization, will be examined. In the past, various means have been devised to search the parameter space for a combination of features that were optimal in describing subtle patterns via high level representation of the data. There is always a direct relationship between deep learning and hyperparameter selection. Therefore, automating the hyperparameter search cannot be trivial as more hyperparameters are often present in the model design. Therefore, prompt location, skillful features learning and better tuning of key hyperparameters to achieve better recognition result in Human Activity Recognition (HAR) are key steps that will be discussed in this chapter. Again, the Bayesian optimization process will be considered for hyperparameter space search for optimal solution. A better optimal hyperparameter combination and accurate regularization method can become a panacea for obtaining a state-of-art recognition model for HAR. Although adequate attempts have been made to elucidate all the findings in this research, it is established that deep learning findings can vary from one model to another.

5.2 Architectural Design of Convolutional Neural Network

In CNNs, layers of convolution, subsampling and discretional fully connected layers are the building blocks of this architecture as described in Figure 5.1. CNNs perform well at describing and encoding important features in images, videos and objects. Such features are explicitly used for recognition and learning purposes. Therefore, to extract discriminatory features from our data, a deeply connected CNN model will be constructed with multilayered neural network as described in section 5.1. The composition of these multilayered networks is convolution, activation function, pooling, dense layer, softmax loss process and the output. Each of these layers will be examined to enable a better understanding of CNNs and their deep model compositions. The aggregate features extracted from the different layers make this method most unique and distinct in their power of recognition.

CNN's advantage over conventional neural network makes it an architecture of choice for human activity recognition. Firstly, CNNs can leverage and circumvent the use of numerous parameters, a common characteristic peculiar to conventional neural

5.2 Architectural Design of Convolutional Neural Network

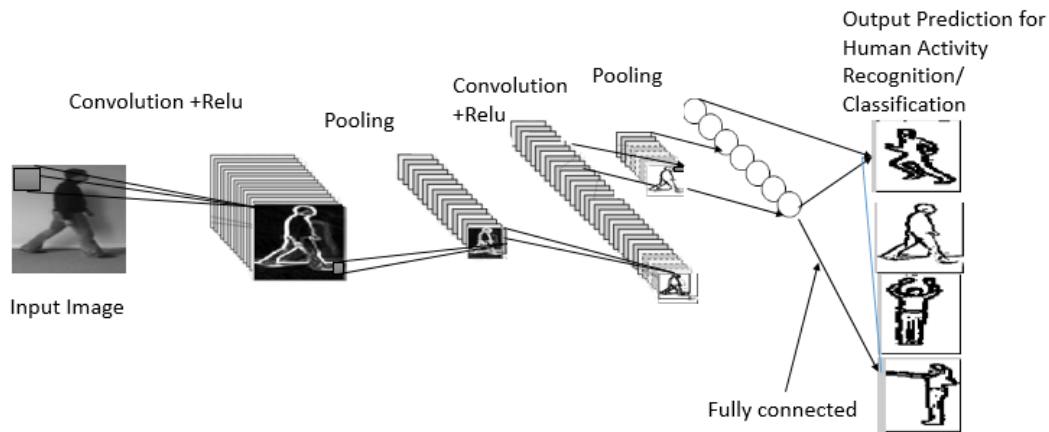


Figure 5.1: Graphical Representation of Deep Learning Process - Convolutional Neural Network Architecture Design for Human Activity Recognition

network. This is made possible with the spatial mapping of neurons from one layer to another. This provides the robustness of controlled parameters compared to the fully-connected neurons found in regular neural network. Secondly, CNNs employ the phenomenon of 3D volume of neurons such that the local patches in the input image are represented in 3 dimensions: width, height and depth. Depth represents the numbers of colour channels. Additionally, CNNs have a final output layer $1 \times 1 \times n$, because the end of CNNs architecture presents a single vector of class scores. n represents the number of classes present for purposes of prediction.

5.2.1 Convolution Layer

The convolutional layer is the kernel responsible for heavy computational processes in a convolutional network. The main concept of convolution is to enable feature extraction from input image. In order to preserve the spatial correlation that is common among the pixel values in an image, a learning process called convolution is performed on the input image. This is achieved by sliding learnable filters across the input image. As the filters slide along the three dimension of the input image, a dot product is computed between the input pixel values and the filters. A 2-dimensional activation map is formed in sympathy and response to the filters spatial relationship with the image pixels positions. These activation volumes formed from different learnable filters

5.2 Architectural Design of Convolutional Neural Network

can encode specific features from the input image, which are stacked along the image depth dimension to form output volumes.

5.2.2 Locally Connected Network

This section highlights how the local region connectivity is utilized. This phenomenon is the restriction of neuron connection to a fixed and small part of the input volume (unit). The spatial dimension of such local connections to the input image forms the hyperparameter known as the receptive field of the neurons. Each neuron receptive field dimension is often equally the same with the filter size applied to the input image. While the local space inherits dimensions of the filter size (height and width) its depth is always equal to the depth of the input image (volume). Image depths that are of RGB characteristics will certainly have a depth of three, and one, for grayscale images. Importantly, it will be crucial to mention that restriction of neurons to local space on the input volumes are only for height and width; whereas input volume depth is fully represented from one end to another. Furthermore, suppose the dimension for a typical RGB input volume size is given as $[64 \times 64 \times 3]$ and if a receptive field of each neurons is chosen as 3×3 , then the convolution layer can attain a weight to the tune of $[3 \times 3 \times 3]$ mapped to its local input region. The weight will, therefore, be $3 \times 3 \times 3 = 27$ weights (and +1 bias parameter). Again, considering a grayscale input image with dimension $[32 \times 32 \times 1]$ and a receptive field of 5×5 , the total weight contribution towards the local region of grayscale image will be $5 \times 5 \times 1 = 25$ weights (and 1 bias parameter). The connectivity in depth for the RGB and gray scale image will, therefore, become 3 and 1 respectively, as these called input volume depths.

5.2.3 Spatial arrangement of output volume

Output volume is a product of neuron's manipulation, just as input volume is determined by the mapping of neurons to local input images. Output volume size is influenced by three major hyperparameters: the depth, stride and zero-padding.

1. Depth - The depth is a hyperparameter that makes up the output volume and this is equivalent to the number of filters to be used. The numbers of filters chosen is responsible for the depth of each activation map. The unique characteristics

5.2 Architectural Design of Convolutional Neural Network

of these filters are their ability to recognize and activate neurons along the depth dimension when certain patterns are found in the input image.

2. Stride - The displacement of filters in relation to the number of image pixels as they are moved round input volumes is called the stride. A slide of filter that amounts to the move of one pixel at a time is called a stride of 1. However, with an uncommon stride of 2 or 3, filters move 2 or 3 pixels at a goal as they slide along the input volume. The use of 2 or 3 stride results in smaller output volumes.
3. Zero padding - This is another hyperparameter that is used for controlling output volume spatial size. In preserving output volume size, it is necessary to pad the input volumes with some number of zeros about the border. Such zero-padding helps retain the spatial size of input volume in a way that both input and output volume dimensions.

The spatial size of the output volume can be computed by a simple formula as shown in Equation (5.1). This formula comprises of input volume (W), receptive field size (F), the number of strides applied (S) and the number of zero-padding used (P). The spatial size of the output volume can, therefore be computed as:

$$\frac{W - F - 2P}{S} + 1 \quad (5.1)$$

Therefore, for an input volume of 5x5, a receptive field size of 3x3 with a stride of 1 and with a no zero padding produces an output volume size of 3x3. The choice of having a stride of 2 in this case results in an output volume size of 2x2.

5.2.4 Rectified Unit

Regardless of the convolutional neural network depth, the rectified unit layer is a principal characteristic of most deep learning models. The use of such non-saturated activation function (ReLU) over other saturated (sigmoid and tanh) functions has been predominately prominent in most deep learning systems. This is because of their effectiveness in solving most of the problems that are peculiar to training deep neural networks. Non-saturated activation function helps the deep learning network in two major areas. Firstly, it helps to solve the negative effect of exploding/vanishing gradient. Secondly, it improves the convergence speed which is very important but difficult

5.2 Architectural Design of Convolutional Neural Network

to realize in deep neural networks. Examples of non-saturated rectified unit are the standard rectified linear unit (ReLU), parametric rectified linear units (PReLU), leaky rectified linear unit (Leaky ReLU), and the recent randomized leaky rectified units (RReLU). For every backpropagation performed, there is a desire to adjust and tune the gradient to minimize the cost function. These constituents that define gradient update are made up of several factors, namely, the derivatives of weight, activation function and biases. During gradient upgrade, as these derivatives travel down each layer, they are propagated away from the output layer towards the input layer. Thus, much multiplicative complexity is introduced. If the multiplication aggregates of such derivatives amount to less than 1, then such aggregates tend towards zero and the vanishing gradient problem is likely to occur. On the contrary, with multiplicative derivatives aggregates greater than 1, such aggregates build up towards infinity resulting in an exploding gradient as the derivatives move closer to the input layer. The RELU seems to fix this problem because they have a gradient of 1 when multiplicative output ≥ 0 , and zero on the contrary. Therefore, the RELU function somewhat creates a means to getting all the derivative of the activation function to be one, thereby preventing the problems of vanishing or exploding gradient. The rectified linear unit is by far the most prominent one that is being used [164, 188]. A rectified linear unit layer performs thresholding that allows input negative or smaller values less than zero is pruned to zero, while retaining values that are greater than zero. Such evaluation as seen in Equation 5.2 depicts the mathematical expression of rectified linear unit, while its derivative is represented in Equation 5.3.

$$F(x) = \max(x, 0) \quad (5.2)$$

$$f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (5.3)$$

It is widely believed that the exceptional performance of ReLU is as a result of the sparse activation occasioned by passing ReLU [163, 188]. Though ReLU has been widely voted for its performance, regrettably it also has its drawbacks. ReLU units tend to be very weak when training and can easily lose track of its activation function and die. The horizontal line as shown in Figure 5.2 is common in ReLU. This tends to gravitate gradients values toward zero mark, thereby causing large gradients streaming

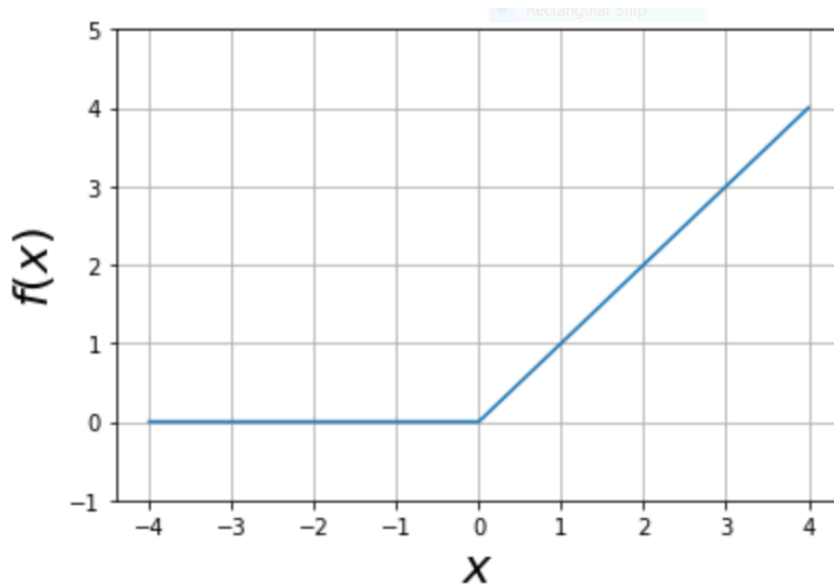


Figure 5.2: RELU Diagram - RELU Graphical Representation

through a ReLU region to stop updating. The neurons in this region become dead to error propagated for training because the gradient is stale and becomes unresponsive to such tuning during training. This is known as the dying ReLU problems. In recent times, the leaky ReLU unit and their variants have been considered as better rectified function that can help solve the difficulties experienced in using the ReLU. In contrast to the former, the leaky ReLU replaces the horizontal line with a slightly inclined line as opposed to the horizontal line of the traditional ReLU. The concept of this method is to stop a zero-gradient function and instead allow the gradient to recover from training large amount of data

5.2.5 Sigmoid Activation Function

The sigmoid activation function is another non-linear activation function that is commonly used in neural network design and training. It can produce analogue activation unlike the step function character that is associated with the logistic regression model. Great and smooth gradients can be derived from the region between the two-flat side. As seen in Figure 5.3, a small change in the X value bring about a significant change in the Y values. With the ability to influence activation to move either side of the curve,

5.2 Architectural Design of Convolutional Neural Network

it is generally considered a choice activation function in many classification problems. Equations 5.4 and 5.5 represent the mathematical representation and the derivatives of the sigmoid activation function while Figure 5.3 shows the diagrammatic representation. The Sigmoid output activation function is squashed between (0,1) and this prevents an activation blow up, which is a phenomenon often experienced in the linear function. Although the sigmoid activation is very useful in most learning algorithms, it has downsides. The vanishing gradient problem is common in this kind of activation function because towards the flat end of the sigmoid curve, the Y values gradually become unresponsive to changes in X. With this concept, learning in the network is extremely difficult as the gradient cannot evolve new changes because it is very small.

$$\text{sig}(A) = \frac{1}{1 + e^{-x}} \quad (5.4)$$

$$\frac{\partial}{\partial t} \text{sig}(A) = \text{sig}(A)(1 - \text{sig}(A)) \quad (5.5)$$

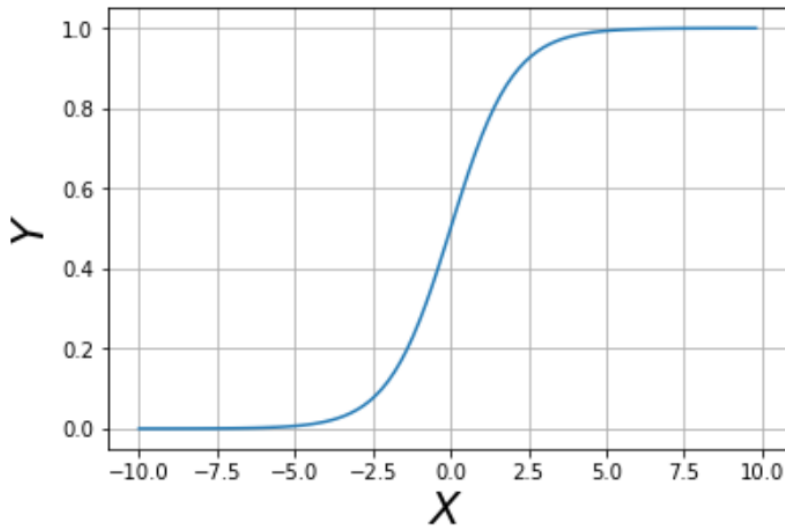


Figure 5.3: Sigmoid Diagram - Sigmoid Activation Graphical Representation

5.2.6 Tanh Activation Function

The tanh activation function is another form by which nonlinearity is introduced into neural networks. It is very similar to the sigmoid activation function. The tanh function squashes output activation between $(-1, 1)$, thus making negative values coming from neural network input to the tanh function to be mapped to negative output values. Also, all the zero-valued tanh inputs output near-zero values. These characteristic properties peculiar to the tanh activation function makes the neural network less likely to behave unresponsively to changes applied to the X values as seen in Figure 5.4. The Tanh function is defined as

$$F(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{2}{1 + e^{-2x}} \quad (5.6)$$

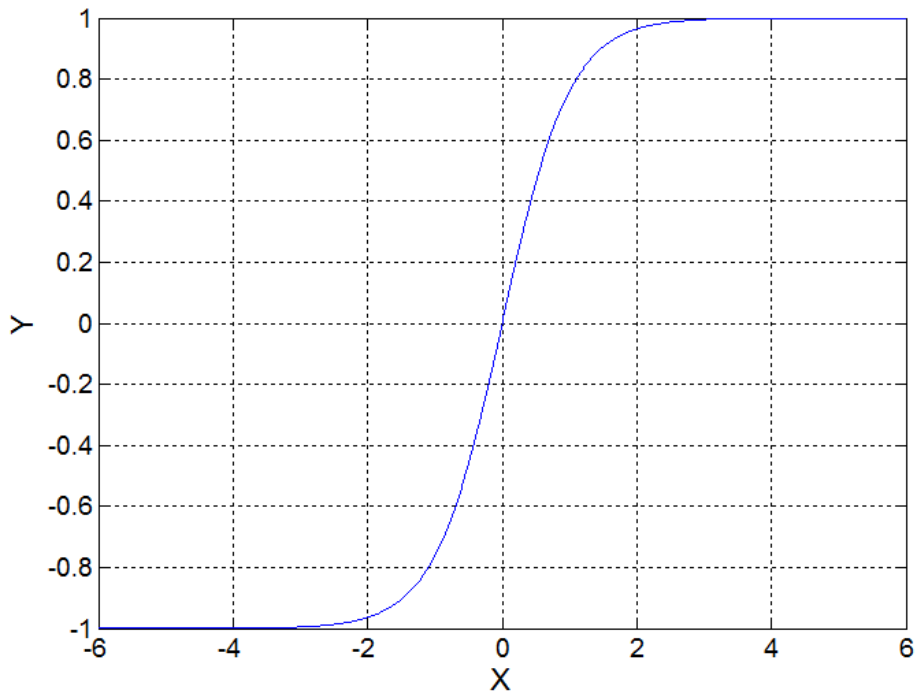


Figure 5.4: Tanh Diagram - Tanh Activation Graphical Representation

5.2.7 Pooling Layer

In CNNs, the features and total number of the parameters produced after each convolution layer can exponentially grow large and this can become a real difficult task

5.3 Deep Learning Class Discrimination in Human Activity Recognition

because of the computational burden on the classifier. Such a model, at the very least, experiences overfitting and generalizes poorly. Therefore, a common practice that can leverage these challenges in the deep learning network is to periodically introduce a pooling layer within consecutive convolutional layers. These pooling (non-linear down-sampling) layers are progressively used for down-sampling the output volumes which discard non-maximal feature values leading to a smaller output volume size. The spatial size obtained from the smaller output volume are a computational gain for the immediate upper layer. Furthermore, a significant reduction in the numbers of parameters used for training is achieved thereby reducing the effect of overfitting in the deep learning model. The pooling layer using a MAX operation independently influences and resizes the dimensions of every output volume along its width and height while preserving the depth slice. Max pooling splits each output volumes into regions of non-overlapping squares and the maximum value on each square sub-region is chosen to represent each partitioned square sub-region. Most of the pooling layers for down-sampling have filter sizes of 2x2 and slides of 2. This produces half of the original volume in width and height while preserving the depth of the final output volume.

5.2.8 The Fully Connected Layer

The transformation of the input images through layers of convolution and pooling has enabled discriminant features to be extracted. However, high-level reasoning around such extracted features by the deep neural network is needed to achieve a proper recognition model built with CNNs architecture. This high-level reasoning is made possible by the fully connected layers, which are the last stage of the CNNs. This layer is fully connected to the previous layer and the features obtained are flattened and can be visually seen as one-dimensional vector. The output uses this one-dimensional vector to formulate various classes present in the network.

5.3 Deep Learning Class Discrimination in Human Activity Recognition

A conventional overview of the CNNs architecture is shown in Figure 5.1. It depicts a transformation of the input image through a phase of a series of convolution, non-linear activation, pooling and fully connected layers to get an output label or a probability

5.3 Deep Learning Class Discrimination in Human Activity Recognition

function that best describes the label of the desired image. This last layer is fully connected to the previous layer and the features obtained are flattened and can be visually seen as one-dimensional vector. The output uses this one-dimensional vector to formulate inferences about the likelihood of a score of various class labels at the network output. With CNNs, the softmax loss is used as the traditional loss function. This loss function allows for deep features of distinct classes to be separated and effect training of any deep neural network.

5.3.1 Challenges of Deep learning Class Discrimination with CNN

The transformation of the input images through layers of convolution and pooling has enabled discriminant features to be extracted. However, high-level reasoning around such extracted features is needed to achieve proper recognition[37, 62, 79, 189]. Softmax activation and cross-entropy loss function ensure the high-level reasoning seen in the fully connected layers. These two functions allow for the probabilistic interpretation of the models ability to learn from the entire datasets. Due to the structural complexity of human actions, the deeply learned features are highly required to possess separable and discriminative power ability. Such discriminative power allows for effective class distinction, generalization of unseen data and thus improves quick convergence of the learning model. It is desirable for the human activity recognition model to seek compact within-class variation and a separable between-class difference, which are only realizable with powerful discriminative features. In conventional CNNs, the softmax loss which is the most prominently known constituent of logit layer is known for its primary role of aiding the separability of features. In [189], the performance of the softmax was reported to lack sufficiency in their discriminative power, prompting the authors to introduce the concept of joint supervisory loss function. The discriminative learning in CNNs can become very difficult because the optimization process in CNNs is often done with the stochastic gradient descent (SGD). The SGD does its optimization processes with mini-batch evaluations. This optimization process suffers from poor global distribution of deep features. The challenges introduce a non-trivial process of obtaining a true and efficient loss function in CNNs. The contrastive loss [190] and triplet loss [191] used in the recognition of the image field attempted to circumvent these shortcomings by using loss functions which use image pairs and triplets. Attempting to use all input images for training purposes is a major disadvantage of

5.4 Loss Function and Convolutional Neural Networks Optimization

these two processes. Other disadvantages with the use of both contrastive and triplet losses are that they grow in dimensions exponentially, converge slowly towards optimum parameters and have huge computational burden to the learning system. In [189], a novel loss function called center loss was introduced to efficiently improve the discriminative influence of the features coming from a deeply learned convolutional neural networks. This method first learns a center for each class present in the training set and the second stage is to penalize the difference between the deep features and their corresponding class centers. During model training, the model attempts to minimize the distance between the deep features and their class centers while equally providing update for the class center. This center loss in conjunction with the popular softmax loss helps to supervise learning and training of the CNNs. The hyperparameter introduced is used in fine-tuning the two supervision signals. The responsibility of keeping deep features of different classes apart is handled by the softmax loss while the center loss function maintains intra class compactness within the same class. The center loss can achieve this through its center pulling property of deep feature from the same class. With this kind of dual combination of different loss function, the proposed joint supervisory model was able to achieve a record state-of-the-art in face recognition[189]. Although this method has been widely adopted and modified for face recognition, we have taken this idea further into the human activity recognition space domain. To the best of our knowledge, no such discussion is in the research domain of HAR. The experimental works on center loss conducted on HAR show that this method performed poorly to create powerful discriminative deep learning features for efficient HAR. This is partly due to challenges and complexities inherent in the body pose structures that are familiar in actions performed by humans.

Therefore, to improve the functionality of the center loss in the joint supervisory learning proposed in [189] and have it made more effective in HAR, we have introduced a new regularization method called likelihood prior probability.

5.4 Loss Function and Convolutional Neural Networks Optimization

The outstanding property of the softmax function to present the layers output in the form of a probability distribution contributes to CNNs learning ability. It is often used

5.4 Loss Function and Convolutional Neural Networks Optimization

as the last layer for most deep neural networks. This process creates visual and metric score calculation that is needed in evaluating the reliability of the learning model. Therefore, obtaining the loss at the end of any deep neural network helps to calculate the gradient. Propagating such gradient backward to the previous layer ensures model optimization[93, 144, 163, 167]. In this section, we elaborate our discussion on loss function and its useful learning achieved through backpropagation of errors. The back propagation method has been around for a long time. It is a fundamental process known for learning, fine-tuning and optimization of neural networks in artificial intelligence and machine learning. The underlying principle of a function approximator like the neural network is to propagate data through different layers of the networks until the last layer has been reached. The last layer is called the output where network predictions are evaluated. In computing the neural networks efficiency, a cost function metrics is defined. The cost function E as depicted in Equation (5.7) is the discrepancy between the targeted class t_T and the output label prediction O_T^l of the function approximator.

$$E = \frac{1}{T} \sum_{T=1}^T (t_T - O_T^l) \quad (5.7)$$

The primary aim of the training process is to make this cost function as low as possible. To achieve this, the gradient of the cost function is computed by a repetitive application of the chain rule principle. The method of recursively expressing the gradient of the error or cost function such that a global minimal is reached is called the gradient descent method of optimization. The output error derivative is calculated with respect to every weight represented in the network and the iterative traversing of this error back to a convex optimization algorithm using the chain rule is popularly known as backpropagation. With this optimization method, neural networks can improve their prediction metric due to neurons weight adjustments achieved through back propagation. The two major update attributes expressed by back propagation in CNNs are the weight and deltas (error difference). These are core factors in neural networks. Their mathematical formulation can be expressed as follows

1. We assume that l will represent the l^{th} layer, $l = 1$ is the first layer and $l = L$ represent the last layer.
2. Let x , $i \times j$, and $H \times W$ be the input, iterator and its dimension.

5.4 Loss Function and Convolutional Neural Networks Optimization

3. The filter w is of size $R_1 \times R_2$ and has an iterator $m \times n$.
4. Let $w_{m,n}^l$ be the weighted vector connecting the neurons in layer l to the layer $l - 1$.
5. Let the bias at layer l be b^l
6. The convolved input vector at layer l given as $x_{i,j}^l$ plus a bias is given as

$$x_{i,j}^l = \sum_m \sum_n w_{m,n}^l O_{i+m,j+n}^{l-1} + b^l \quad (5.8)$$

7. $O_{i,j}^l$ represents the output vector at the layer l , formulated from

$$O_{i,j}^l = f(x_{i,j}^l) \quad (5.9)$$

8. The activation function is $F(\cdot)$, and the activation of the convolved input is represented as $F(x_{i,j}^l)$.

A computation to deduce the gradient which interprets the degree of change of a single pixel $w_{m',n'}$ between the weighted kernel and the error function is given as

$$\delta_{m',n'}^l = \frac{\partial E}{\partial w_{m',n'}^l}.$$

If we convolve the input features map of size $H \times W$ with a kernel weight of size $R_1 \times R_2$, an output feature map of size $(H - R_1 + 1) \times (W - R_2 + 1)$ is realized. Exploring the gradient of each individual weight can be achieved by applying the chain rule given in Equations 5.10 through 5.13

$$\begin{aligned} \frac{\partial E}{\partial w_{m',n'}^l} &= \sum_{i=0}^{H-R_1} \sum_{j=0}^{W-R_2} \frac{\partial E}{\partial x_{i,j}^l} \frac{\partial x_{i,j}^l}{\partial w_{m',n'}^l} \\ &= \sum_{i=0}^{H-R_1} \sum_{j=0}^{W-R_2} \frac{\partial E}{\partial x_{i,j}^l} \frac{\partial x_{i,j}^l}{\partial w_{m',n'}^l} \end{aligned} \quad (5.10)$$

Recall that from Equation (5.8) that $x_{i,j}^l$ is given as $\sum_m \sum_n w_{m,n}^l O_{i+m,j+n}^{l-1} + b^l$. When this part is further expanded,

$$\begin{aligned} &\frac{\partial x_{i,j}^l}{\partial w_{m',n'}^l} \\ &= \frac{\partial}{\partial w_{m',n'}^l} \left(\sum_m \sum_n w_{m,n}^l O_{i+m,j+n}^{l-1} + b^l \right) \end{aligned} \quad (5.11)$$

5.4 Loss Function and Convolutional Neural Networks Optimization

Expanding and further expressing the partial derivatives in Equation (5.11) will amount in zero values for all the components except for those where $m = m'$ and, $n = n'$ in $w_{m,n}^l O_{i+m,j+n}^{l-1}$

$$\begin{aligned} & \frac{\partial x_{i,j}^l}{\partial w_{m',n'}^l} \\ &= \frac{\partial}{\partial w_{m',n'}^l} (w_{0,0}^l O_{i+0,j+0}^{l-1} + \dots + w_{m,n}^l O_{i+m,j+n}^{l-1} + \dots + b^l) \end{aligned} \quad (5.12)$$

$$\begin{aligned} &= \frac{\partial}{\partial w_{m',n'}^l} (w_{m,n}^l O_{i+m,j+n}^{l-1}) = O_{i+m,j+n}^{l-1} \\ & \frac{\partial E}{\partial w_{m',n'}^l} = \sum_{i=0}^{H-R_1} \sum_{j=0}^{W-R_2} \delta_{i,j}^l O_{i+m,j+n}^{l-1} = \text{rot}_{180}^0 \{ \delta_{i,j}^l \} * O_{m,n}^{l-1} \end{aligned} \quad (5.13)$$

Substituting Equation (5.12) in equation (5.10), we obtain the following:

The double aggregate as seen in Equation (5.13) stems from the weight sharing process in the network as the same weight kernel is convolved across all the local receptive fields in the input feature map. This summation depicts the pooling of each and every gradients $\delta_{i,j}^l$ originating from all the output layer l . The derivation of the gradient in relation to the filter maps can also be deduced as a transformation phase, (i.e cross-correlation to convolution), visibly apparent in the flipping of the delta matrix represented in the right hand side of Equation (5.13) as rot_{180}^0

The gradient $\delta_{i,j}^l = \frac{\partial E}{\partial x_{i',j'}^l}$ computed from the error change or loss function E with respect to the pixel changes $x_{i',j'}^l$ in the input feature map can also be derived using the chain rule from the following equation:

$$\frac{\partial E}{\partial x_{i',j'}^l} = \sum_{i,j \in V} \frac{\partial E}{\partial x_V^{l+1}} \frac{\partial x_V^{l+1}}{x_{i',j'}^l} \quad (5.14)$$

The input pixel $x_{i',j'}^l$ influencing the output region can transverse from the top left corner of the input. The effect of the input pixels $x_{i',j'}^l$ on the output region can transverse from the top left corner $(i' - R_1 + 1, j' - R_2 + 1)$ to the bottom right corner (i', j') of the output region.

$$\frac{\partial E}{\partial x_{i',j'}^l} = \sum_{i,j \in V} \delta_V^{l+1} \frac{\partial x_V^{l+1}}{x_{i',j'}^l} \quad (5.15)$$

5.4 Loss Function and Convolutional Neural Networks Optimization

V in Equation (5.15) denotes the output area being influenced by the single pixel $x_{i',j'}^l$ from the input feature map. An explicit representation of this is given in Equations (5.16) and (5.17)

$$\frac{\partial E}{\partial x_{i',j'}^l} = \sum_{m=0}^{R_1-1} \sum_{n=0}^{R_2-1} \frac{\partial E}{\partial x_{i'-m,j'-n}^{l+1}} \frac{\partial x_{i'-m,j'-n}^{l+1}}{\partial x_{i',j'}^l} \quad (5.16)$$

The region V is defined by a height range of $i' - 0$ through $i' - (R_1 - 1)$ and having width of $j' - 0$ through $j' - (R_2 - 1)$. The duo as analyzed can further be represented in the summation as $i' - m$ and $j' - n$ with m and n as the iterators of range $0 \leq m \leq R_1 - 1$ and $0 \leq n \leq R_2 - 1$ respectively.

$$\frac{\partial E}{\partial x_{i',j'}^l} = \sum_{m=0}^{R_1-1} \sum_{n=0}^{R_2-1} \delta x_{i'-m,j'-n}^{l+1} \frac{\partial x_{i'-m,j'-n}^{l+1}}{\partial x_{i',j'}^l} \quad (5.17)$$

Recall that from Equation (5.17), $x_{i'-m,j'-n}^{l+1}$ is equal to $w_{m,n'}^{l+1} O_{i-m+m',i-n+n'}^l + b^{l+1}$ and an expansion of this formulae results in Equations (5.18) and (5.19) below:

$$\frac{\partial x_{i'-m,j'-n}^{l+1}}{\partial x_{i',j'}^l} = \frac{\partial}{\partial x_{i',j'}^l} \left(\sum_{m'} \sum_{n'} w_{m',n'}^{l+1} O_{i-m+m',i-n+n'}^l + b^{l+1} \right) \quad (5.18)$$

$$\frac{\partial x_{i'-m,j'-n}^{l+1}}{\partial x_{i',j'}^l} = \frac{\partial}{\partial x_{i',j'}^l} \left(\sum_{m'} \sum_{n'} w_{m',n'}^{l+1} f(x_{i-m+m',i-n+n'}^l) + b^{l+1} \right) \quad (5.19)$$

$$\begin{aligned} \frac{\partial x_{i'-m,j'-n}^{l+1}}{\partial x_{i',j'}^l} &= \frac{\partial}{\partial x_{i',j'}^l} w_{m',n'}^{l+1} f(x_{i-m+m',i-n+n'}^l) \\ &+ \dots (w_{m,n}^{l+1} f(x_{i',j'}^l) + \dots + b^{l+1}) \end{aligned} \quad (5.20)$$

If we follow through Equation (5.17) and take the partial derivatives of all its associated, the components will invariably set each of them to have a zero value, with the exception of components $m' = m$ and $n' = n$. Therefore, with these components, we can equate $f(x_{i-m+m',i-n+n'}^l)$ to become $f(x_{i',j'}^l)$, while $w_{m',n'}^{l+1}$ becomes $(w_{m,n}^{l+1})$

$$\begin{aligned}
 \frac{\partial x_{i',j'}^{l+1}}{\partial x_{i',j'}^l} &= \frac{\partial}{\partial x_{i',j'}^l} (w_{m,n}^{l+1} f(x_{i',j'}^l)) \\
 &= w_{m,n}^{l+1} \frac{\partial}{\partial (x_{i',j'}^l)} (f(x_{i',j'}^l)) \\
 &= w_{m,n}^{l+1} f'(x_{i',j'}^l)
 \end{aligned} \tag{5.21}$$

Conversely, replacing Equation (5.17) with (5.21) will give:

$$\frac{\partial E}{\partial x_{i',j'}^l} = \sum_{m=0}^{R_1-1} \sum_{n=0}^{R_2-1} \delta_{i'-m,j'-n}^{l+1} w_{m,n}^{l+1} f'(x_{i',j'}^l) \tag{5.22}$$

The flipped kernel function in the backpropagation mode is the expression of convolution as flipped kernel cross-correlation. This is seen in Equation (5.23).

$$\begin{aligned}
 \frac{\partial E}{\partial x_{i',j'}^l} &= \sum_{m=0}^{R_1-1} \sum_{n=0}^{R_2-1} \delta_{i'-m,j'-n}^{l+1} w_{m,n}^{l+1} f'(x_{i',j'}^l) \\
 &= \text{rot}_{180^\circ} \left\{ \sum_{m=0}^{R_1-1} \sum_{n=0}^{R_2-1} \delta_{i'-m,j'-n}^{l+1} w_{m,n}^{l+1} \right\} f'(x_{i',j'}^l) \\
 &= \delta_{i',j'}^{l+1} * \text{rot}_{180^\circ} \left\{ w_{m,n}^{l+1} \right\} f'(x_{i',j'}^l)
 \end{aligned} \tag{5.23}$$

5.4.1 Joint Supervisory Loss

The supervision of deep neural network under the softmax loss accounts for a separable feature for HAR classification. However, these features are not discriminative enough to cause a large inter-class variation. Based on this reason, authors in [189] proposed center loss function that will improve the deep feature discriminative ability in neural networks. In [189], the authors demonstrated the key concept behind center loss. The intra-class distance minimization associated with the center loss is a fundamental, unique property that can improve the classification process. A state-of-the-art result was recorded with face dataset. However as seen in Figure 5.6, such fantastic results were not realizable with human activity recognition. The center loss function is given in Equation (5.24)

$$L_c = \frac{1}{2} \sum_{i=1}^m \|x_i - c_{yi}\|_2^2 \tag{5.24}$$

5.4 Loss Function and Convolutional Neural Networks Optimization

L_c represents the center loss. m is the number of training samples present in the mini-batch size. The $x_i \in R^d$ denotes the i th deep feature of the y_i class and d is the feature dimension. The $c_{y_i} \in R^d$ denotes the y_i class of the deep features. The softmax loss presented in Equation (5.25) is combined with center loss as described in [189] for training deep neural networks. A combination of both loss function is shown in Equation (5.26). Where $W_j \in R^d$ denotes the j th column of the weights present in the terminal layer and $b \in R^n$ represents the bias term. The symbol n represents the number of classes in the training data.

$$L_s = \sum_{i=1}^m \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}} \quad (5.25)$$

$$L = - \sum_{i=1}^m \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^m \|x_i - c_{y_i}\|_2^2 \quad (5.26)$$

$$L = L_s + \lambda L_c \quad (5.27)$$

The overall loss for the deep neural network is denoted by L , while the softmax and center loss are denoted by L_s and L_c respectively. The λ is a scalar for fine tuning the two loss functions.

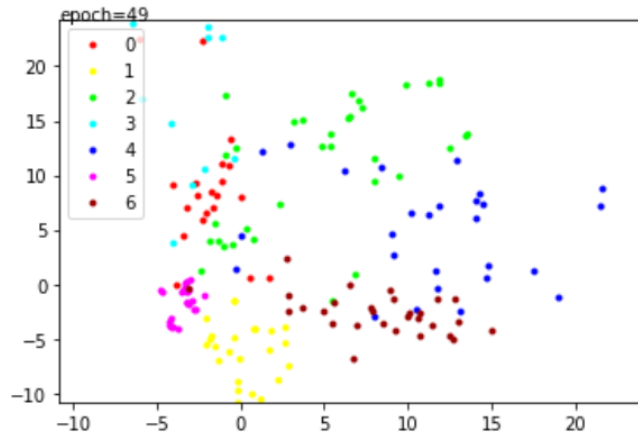


Figure 5.5: Softmax Loss Graphical Representation - Distribution and visualization of deeply learned features using only softmax loss for Human Activity Recognition

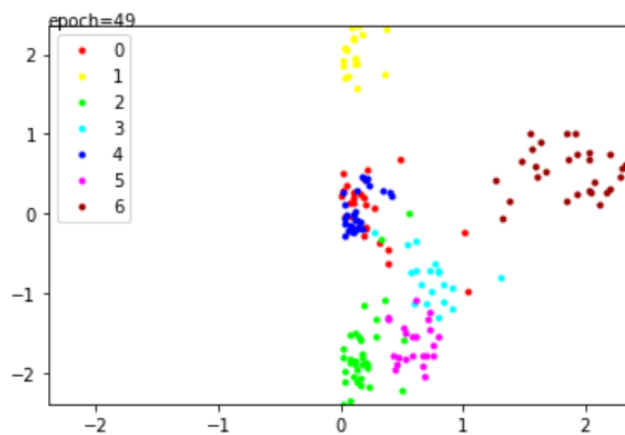


Figure 5.6: Center Loss Graphical Representation - Distribution and visualization of deeply learned features using the joint supervision of softmax and the center loss for Human Activity Recognition.

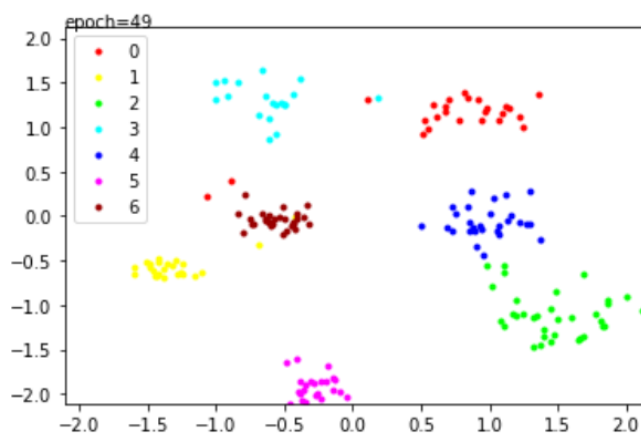


Figure 5.7: Proposed Joint Supervisory Loss Graphical Representation - Distribution and visualization of deeply learned features using our proposed loss for Human Activity Recognition.

5.4.2 Our Proposed Method

While center loss continues to be popular and widely significant in the recognition of face and other visual classifications, its poor class discriminative performance as seen in Figure 5.6 exposes its inadequacy in handling complexities posed by the structure of the human body. A fundamental weakness associated with center loss method is its poor discriminative power between features of different class in HAR. Good and effective discriminative features should have high intra-class compactness and inter-class separability [29, 69, 79, 146, 192].

In Consideration of the foregoing, our proposed approach is to introduce a likelihood regularization term that will allow model parameters to effectively learn processes that improve the achievement of both intra-class compactness and inter-class separability in HAR. In this method, we intuitively mined the domain knowledge presented by the joint probability distribution theory. This concept will further elucidate the relationship between posterior probability of extracted deeply learned features belonging to a given set of class and conventional joint supervisory method.

Equations 5.24, 5.25 and 5.26 represent the euclidean distance between the deeply learned features and their respective class centriod (center loss), softmax loss and the joint supervision of softmax and center loss respectively. A simplification of Equation (5.26) by representing the softmax, center loss with L_s and L_c respectively is shown in Equation (5.27). From Equation (5.27), we can infer that the center loss acts as a regularization term to the softmax loss. Therefore, the process of obtaining better discriminant features from a deep neural network can be likened to the regularization of softmax loss which is a measure of the score probability. Considering the softmax loss as described in Equation (5.25), the input sample x , which represents the extracted deep feature vector can be described in terms of posterior probability.

The posterior probability of x that can be present in a particular class $y \in [1, K]$ is described in Equation (5.28). The logit, a measure of the score $f_k(x)$, is a linear transformation of the feature vector x as shown in Equation (5.29). A higher score from the linear combination of all the weights w and biases b from Equation (5.29) reflects better posterior probability of the likelihood of x being part of class k .

$$p(y/x) = \frac{e^{f_y(x)}}{\sum_{k=1}^K e^{f_k(x)}} = \sum_{i=1}^m \log \frac{e^{W_y^T x_i + b_y}}{\sum_{k=1}^K e^{W_k^T x_i + b_k}} \quad (5.28)$$

5.4 Loss Function and Convolutional Neural Networks Optimization

$$f_k(x) = W_k^T x_i + b_k, k \in [1, K] \quad (5.29)$$

From the observation of Equation (5.27), we can logically assert that the Euclidean distance, which represents the distance between the extracted features and class centroid is acting as a regularization term for the softmax loss function. This method has gained popularity in the recognition of face classification[4, 189, 193], but such gain as seen in Fig.5.6 are too small for a complex model like human recognition. Thus a complex model will need an adequate regularization term that will punish heavily weighted parameters such that a smooth gradient is realized.

$$p(x) = \sum_{k=1}^K N(x; \mu_k, \Sigma_k)p(k) \quad (5.30)$$

In this work, we aim to introduce an additional prior distribution regularization term to even out the effect of high model complexity that impede better discrimination amongst the different class and the mechanism of obtaining a well generalized inference model. A notable assumption made by the authors in [193], is that extracted deep feature x obtained from the model will be modelled as a Gaussian mixture distribution(GMM) shown in Equation (5.30). From Equation (5.30), the prior probability is $p(k)$ in class k , with μ_k and Σ_k being the mean and covariance of a given class k respectively.

The most popular forms of inference that GMM relies on are density estimation and clustering and these two factors strongly correlate the work done in [189]. The distribution and visualization of deeply learned features as seen in Fig.5.5 and Fig.5.6 describe a bias toward clustering inference where each colour represents different deep features clusters.

Considering the general assumption made, the likelihood probability of feature x_i with a known class $z_i \in [1, K]$ can duely be repressed as shown in Equation (5.31), while a posterior probability distribution can be described in Equation (5.32).

$$p(x_i | z_i) = N(x; \mu_{z_i}, \Sigma_{z_i}) \quad (5.31)$$

$$p(z_i | x_i) = \frac{N(x; \mu_{z_i}, \Sigma_{z_i})p(z_i)}{\sum_{k=1}^K N(x; \mu_k, \Sigma_k)p(k)} \quad (5.32)$$

5.4 Loss Function and Convolutional Neural Networks Optimization

Similarly, it is also possible to approximate the posterior component assignment using Maximum-a-Posterior (MAP), Bayes' theorem and estimated model parameters. In [194], the MAP predictors which are used for learning structured label problems, a posterior distribution paradigm that can be fine tuned by stochastic gradient descent over perturbation range are developed. Observations from this work suggest that: assuming a training datasets of occurrences and target labels, the learning problem can be the approximation of the parameters of the learning model. This learning model help define subsequent labels detected by each unique instance. The loss function is used for evaluating the fitness of each instance.

Based on the Bayes rules as seen in Equation (5.33), mimicking its unique characteristics that allow useful update and distribution about model parameters based on observed data is proposed. This method of maximizing a posterior probability estimate (MAP) plays a major role in the modelling scheme of the joint probability loss function with an adequate prior regularization terms $\frac{\lambda}{\log N(2\pi)}$. To validate the MAP estimate, Equation (5.33) can be written in term of the Log function as shown in Equation (5.34). The regularization term is the prior distribution update parameter that does the inferences on the maximization of the posterior estimate.

$$p(z_i | x_i) = \frac{p(x_i | z_i)p(z_i)}{p(x_i)} \quad (5.33)$$

$$\begin{aligned} z_i^{map} &= \arg \max_{z_i} p(z_i | x_i) \\ &= \arg \max_{z_i} \frac{p(x_i | z_i)p(z_i)}{p(x_i)} \\ &= \arg \max_{z_i} p(z_i)p(x_i | z_i) \\ &= \arg \max_{z_i} \text{Log}(p(x_i | z_i)) + \text{Log}(p(z_i)) \\ &= \arg \max_{z_i} \sum_{n=1}^N \text{Log}(p(x_i | z_i)) + \text{Log}(p(z_i)) \end{aligned} \quad (5.34)$$

In light of the foregoing, the Bayes' theorem in Equation (5.34) and GMM also shown in Equation (5.30) can be used to describe posterior probability distribution of our deeply learned features x and this is because MAP estimation of model input and parameters leads to a regularised solution. The prior probability in each equation can be modelled to produce the likelihood regularizer in the proposed method.

5.5 Experimental and CNN Detailed Setup

Therefore, our proposed loss function, L^P , is a combination of softmax, center loss and a likelihood regularization term as shown in Equation (5.35). This likelihood regularization term provides useful information on the posterior likelihood estimation of class label z_i which is the distribution of features and the predicted label. With the stochastic gradient descent training method of deep neural networks seen in section 5.2.2, the network parameters are now constrained to learn optimizing the intra-class compactness, while maximizing the inter-class distance between the different classes.

$$L^P = - \sum_{i=1}^m \log \frac{e^{W_{yi}^T x_i + b_{yi}}}{\sum_{j=1}^n e^{W_j^T x_i + b_j}} + \frac{1}{2} \sum_{i=1}^m \|x_i - c_{yi}\|_2^2 + \frac{\lambda}{\log N(2\pi)} \quad (5.35)$$

Furthermore, going by the theoretical and empirical similarities presented in Equations 5.30, 5.34 and 5.35, it is easy to draw a correlation on all three methods and their unique property of adding extra log-prior- distribution regularization term. This term incorporates prior knowledge of its estimated parameters and λ as seen in Equation (5.35) is used in fine-tuning the regularization process. In this work, λ can vary between 0.2 and 1. However, 1 was the best possible value that improved the discriminative power in this model. Therefore, this characteristic has provided the opportunity of leveraging the shortcomings experienced by the joint supervision of softmax and center loss in HAR. With our proposed method, CNNs for HAR can have a better supervised loss function that ensures the maximization of inter-class separability and higher intra-class compactibility.

5.5 Experimental and CNN Detailed Setup

The hardware setups for the deep neural network learning involve NVidia GeForce GTX 960M windows machine. It has an Intel(R) core(TM) i7-6700HQ Computer Processing Unit (CPU). The KERAS open source neural network software is used for the implementation of CNNs deep learning experiment. The tensorflow framework was used as a backend for this experiment, with the knowledge that KERAS can function on top of neural network libraries as though it was the tensorflow framework itself. The experimental setup consists of input images from the Weizmann and KTH datasets. The Weizmann dataset contains 10 classes of activities. However, seven classes were

5.5 Experimental and CNN Detailed Setup

considered for this experiment and they are: running (run), bending (bend), jumping jack (jack), skipping (skip), galloping sideways (side), waiving-two hands (wave) and jumping forward-on-two-legs (jump).

A total of nine persons were involved in the performance of these seven activities, five of whom were used for the training purpose while the remaining four were used for testing.

CNN Details : When a model is constructed for both training and test purposes, the input shape of the image is of importance. As seen in Table 5.1, the first layer of the CNNs is a sequential model that primarily depends on the input shape for the first time. However, subsequent layers along the CNNs are capable of automatically resolving and dealing with the inference of the shape. An input shape of 40,80,1 representing width, height and channel number were used respectively. The learning of the model is centered around three arguments: optimization, evaluation metrics and the loss function. However, in this case, the three types of loss function experimented with are made visible in the evaluation layer, but one of these is used at a time. The filter size of the convolutional layers is set to 3×3 , followed by a PRelu non-linear activation unit. The number of feature maps are 32 for the first two convolutional layer and there is feature maps increment by a factor of $\times 2$ for each two subsequent upper layers which are 64 and 128 respectively. The max-pooling grid is set to 2×2 on all pooling layers, while a stride of 1 is maintained through out the entire model architecture. The source code algorithm in Table 5.1 outlays the architectural construct of our CNNs model. It consists of interdependent layers of convolution, activation, max-pooling, dropout, and the fully connected layer which comprises of a flattened and dense output vector. It is important to note that this architectural construct is peculiar to our model. As each feature maps transverse from the input layer to the next layer, the output shape tends to become smaller and this eventually becomes equal to an output vector representing the number of classes in the recognition or classification model as seen in Figure 5.1

5.5.1 Experimental results on Weizmann and KTH dataset

In this result section, our model is evaluated on the Weizmann and KTH datasets, a popular dataset that is well known in HAR. A pictorial example of some Weizmann and KTH datasets sample is shown in Figures 5.8 and 5.9 respectively. The Weizmann

5.5 Experimental and CNN Detailed Setup

Table 5.1: Model Architecture

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='PReLU',
padding='same', inputshape=(40, 80, 1)))
convout1 = Activation('PReLU')
model.add(convout1)
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout2 = Activation('PReLU')
model.add(convout2)
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout3 = Activation('PReLU')
model.add(convout3)
model.add(MaxPooling2D(poolsize=(nbpool, nbpool)))
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout4 = Activation('PReLU')
model.add(convout4)
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout5 = Activation('PReLU')
model.add(convout5)
model.add(MaxPooling2D(poolsize=(nbpool, nbpool)))
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout6 = Activation('PReLU')
model.add(convout6)
model.add(Flatten())
model.add(Dense(128))
model.add(Activation('PReLU'))
model.add(Dense(2))
model.add(PReLU)
model.add(Activation('softmax', 'softmax + centerloss', 'proposedloss'))
model.compile(loss='categorical_crossentropy',
optimizer='adadelta', metrics=["accuracy"])
```

image dataset of 910 images were randomly shuffled into a training set of 728 Images and a testing set of 182 samples images. The processed images were resized to 80×40 pixels to reduce excessive computational burden.

5.5 Experimental and CNN Detailed Setup

With the KTH dataset, the images from the video frames are extracted and pre-processing is applied to the extracted image sequence. The KTH dataset comprises of six different kinds of human actions: walking, jogging, running, boxing, hand waving, and hand clapping. These grayscale images were then cropped to 90×35 pixels. A total of 1200 images were used for the training and 210 images for testing.

The discriminative power for each loss function is shown in Figures 5.5, 5.6 and 5.7. However, from the visualization and distribution of features in our newly proposed regularization method, Figure 5.7 has better discriminative power for HAR than for the two other methods (softmax and jointly supervised softmax and center loss). From Figure 5.7, the class separations are distinct which is a huge progress that underscores the importance of features regularization in deep neural network.

The results in Table 5.2 shows the accuracy of all three methods considered in this study. From this table, our proposed method has shown superior recognition accuracy over the softmax loss function and the jointly supervised loss function as in [189]. Our proposed method has achieved an accuracy of 96.40% on the Weizmann dataset, which is a better performance than the 93.67% and 95.30% achieved by the softmax and jointly supervised loss respectively. The similitude between the results from the Weizmann dataset, and the KTH dataset accounts for the improvement in the accuracy of our proposed likelihood regularized loss function. A 95.20% accuracy is observed in our proposed method which is better compared to 93.0% and 94.90% achieved by the softmax and the jointly supervised loss respectively. Figures 5.10, 5.11 and 5.12 demonstrate that our proposed regularized loss function has much capacity to further improve its learning curve during the training process. This is because, as seen in Figure 5.12, it is evident that both training and test set graphs still exhibit a sign of continuity even past the 200th epoch. However, this is different for softmax loss and the jointly supervised loss seen in Figures 5.10 and 5.11 respectively. There exists a flat region around the 130 epoch along Figures 5.10 and 5.11, which is an indication that the network is unable to learn from further iteration as the gradient may have been stuck in the local optimal region during the back propagation process. From the foregoing, it is evident that the regularization term contributed to better training accuracy and better discriminative power of the features in HAR recognition and classification.

5.5 Experimental and CNN Detailed Setup

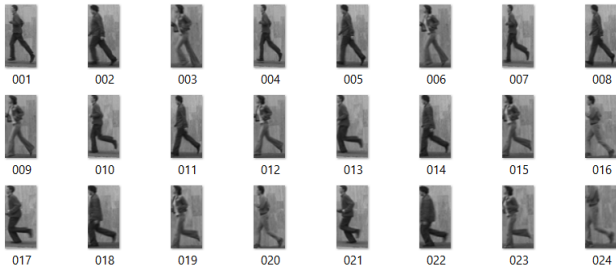


Figure 5.8: Weizman Images - Images of Human activity action from Weizmann video dataset.



Figure 5.9: KTH Images - Images of Human activity action from KTH video dataset.

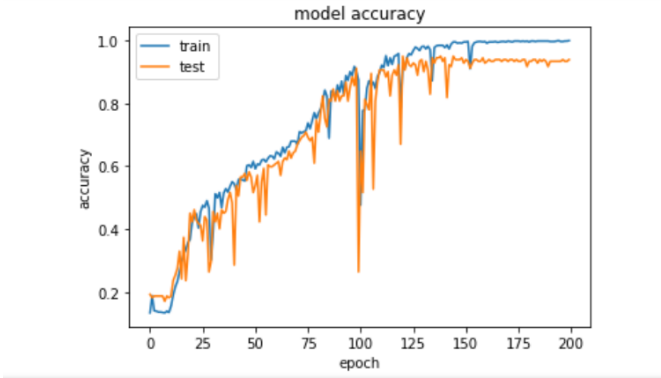


Figure 5.10: Softmax Accuracy - Model accuracy for softmax loss.

5.5 Experimental and CNN Detailed Setup

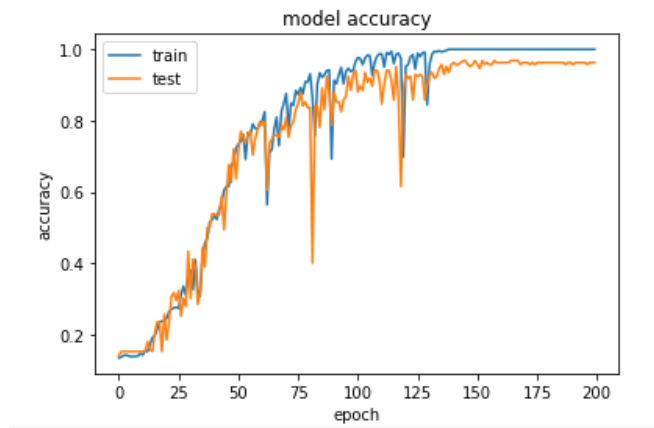


Figure 5.11: Center Loss Accuracy - Model accuracy for center loss.

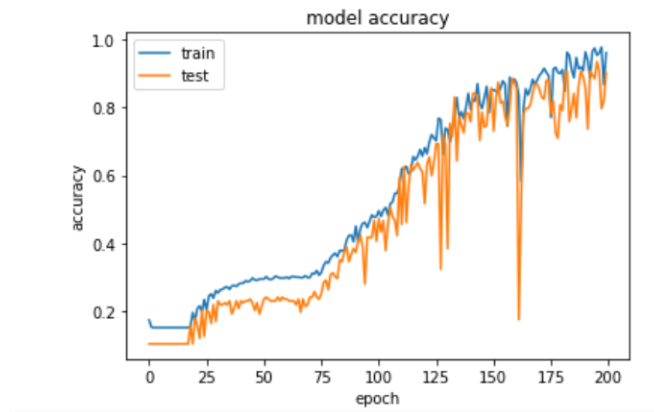


Figure 5.12: Proposed Model Accuracy - Model accuracy for our regularized center loss.

Table 5.2: Accuracy score for different loss function method

Loss Function	Accuracy on Weimann	Accuracy on KTH
Softmax	93.67%	93.0%
Centerloss	95.30%	94.90%
Proposed Method	96.40%	95.20%

5.6 Model Optimization

Model optimization are in different forms and they are necessary for clear feature learning, most of these optimization methods provide significant means of feature regularization. Feature regularization creates means of model expressive characterization and recognitions needed in most computer vision domains and other deep learning areas. The expressive characteristic exhibited by the deep convolutional neural networks model is owed to its ability of mapping a complex relationship between both input and output with the aid of several non-linear hidden units. However, small sample size training data also act disadvantageously by allowing noise presents in the training data to be modeled as a complex relationship. These additional superficial complexities which are adequately modeled in the training sets are not necessarily accounted for in the test sets and this imbalance leads to overfitting. One notable method for overcoming overfitting is to stop the training cycle when it is observed that performance on a validation set starts to loose its strength to generalize on an unseen data. Additionally, L_1 and L_2 weight penalties regularization method have also been considered in Nowlan and Hinton [195]. Another method that was introduced into convolutional neural networks model was the concept of dropout. Dropout has become increasingly popular in its use in deep learning for the optimization and avoidance of unpleasant overfitting. Dropout [196, 197] is a regularization technique that stochastically equates the activation of hidden units to a zero value for every routine training subset per training time. New stochastics averaging model method also contrived by dropout concept were stochastic pooling [198] and Dropconnect [145, 198, 199]. The summary effect of dropout on neural networks is its method to light loose its composition of co-adapted feature detector. This process disables other retained neuron components from

contributing their overall weighting process. Such provisional exclusion of hidden and visible units from the network causes incoming and outgoing weights connections to be dropped as shown in Figure 5.13b. The decision on what units are to be dropped is done randomly and preserving each unit with a fixed probability p which is uninfluenced by other units. In our experiment and for simplicity, we simply set p as 0.5 as empirical results from numerous neural network task have ascertained its optimal characteristic [193]. Interpreted differently, the dropout concept contrives a unique way of averaging modeling trained networks such that the number of trained models is like its individual units at test time, thus allowing the parameters to be shared in such a model. Again, put differently, the application of dropout mechanism is like sampling a thinned neural network model from the original model. Therefore, this thinned network comprises of network units retained after the dropout was done as seen in Figure 5.13b. The possible number of thinned networks from n unit neural networks is given to be 2^n . Weight sharing in this network is common as new thinned networks are trained during each performance cycle. Therefore, applying dropout on neural networks is similar to the training of 2^n thinned networks with maximum weight sharing, also with less training cycle for the thinned networks. The advantage of using dropout is to have multiple simpler co-adapted units which are great at generalizing well with a novel test data rather than a complex co-adaptation from a conventional neural network that is poor on generalization.

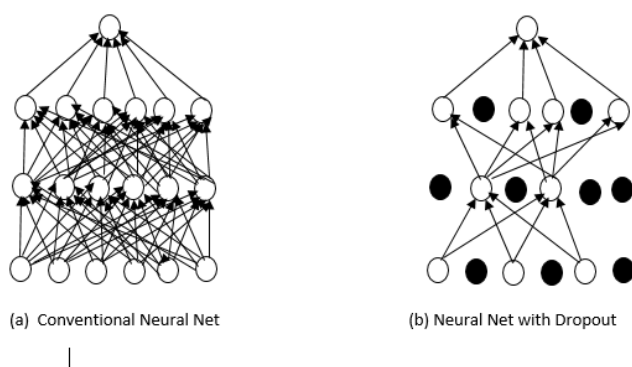


Figure 5.13: Dropout Visualization - Graphical Representation of Neural Networks Dropout

5.6.1 Neural Network Dropout Model Description

In this section, an overview of dropout neural network model is reviewed. Assume a neural network with L numbers of hidden layers, where $l \in \{1, \dots, L\}$ represents the layer index of the neural networks. Let V^l be the input vector going into layer l , while O^l denotes the output vector from layer l . For a conventional fully connected feed-forward networks (*i.e.*) Figure 5.13a with weights w^l , hidden units i , and biases b^l , can be defined as $L \in \{0, \dots, L - 1\}$

$$v_i^{l+1} = w_i^{l+1}O^l + b_i^{l+1} \quad (5.36)$$

$$o_i^{l+1} = f(v_i^{l+1}) \quad (5.37)$$

Suppose f is the activation function given as $f(x) = \frac{1}{(1+\exp(-x))}$
 $r_j^l \sim \text{Bernoulli}(p)$

$$\bar{O}^l = r^L * O^l, \quad (5.38)$$

$$v_i^{l+1} = w_i^{l+1}\bar{O}^l + b_i^{l+1} \quad (5.39)$$

$$o_i^{l+1} = f(v_i^{l+1}) \quad (5.40)$$

The element-wise product is given as $*$, and r^l is the Bernoulli random variable with a probability of p . For the purpose of creating a thinned output network \bar{O}^l , the element-wise multiplication of the sample vector and the outputs of layer O^l are carried out. The Iterative use of the successive thinned output from one layer as the input to another layer amounts to random selection and making a sub-network from a much larger network. The backpropagation method as discussed in section 5.2.2 is used for the learning purpose to ensure that errors are reduced and thus achieve better generalization of our recognition model.

Table 5.3: Key training summary of input parameters to the CNNs

Batch size to train	batch size = 32
Number of output classes	nbclasses = 7
Number of epochs to train	epochs = 35
Number of convolutional filters to use	nbfilters = 32
Size of pooling area for max pooling	nbpool = 2
Convolution kernel size	nbconv = 3

5.6.2 Empirical Results on the Dropout Regularization

The dropout regularization training architecture is slightly modified from that of Table 5.1 to accommodate the dropout layers and other layer parameters that could enhance better optimization of the model. A summary of the parameters of the input training is shown in Table 5.3 and explicitly explains all of the input parameters at the initial stage of our convolutional neural networks. In the experimental setup, a batch-size of 32 was used. The batch-size is the number of training samples that is designed to propagate a forward and backward pass through the neural network architecture in one iteration. An optimal batch-size creates efficient utilization of memory and better training of the model. One such forward and backward pass of a training sample is called the epoch. The visual concept of an epoch is demonstrated in Tables 5.5 and 5.8 and each training cycle is characterized by a cost function which is evaluated by the model to ascertain its accuracy. The backpropagation mechanism of optimization as discussed in Section (5.2.2) provides the next epoch cycle where necessary weight adjustment is needed to further improve model accuracy.

A max-pooling of 2 is considered in the pooling layer of this CNNs model, while the number of convolutional filters and kernel size engaged were 32 and 3 respectively.

A model is constructed for both training and test purposes and the input shape of the image is of importance. The CNNs first layer is a sequential model that primarily depends on the input shape for the first time. However, subsequent layers along the CNNs are capable of automatically resolving and dealing with shape inference. An input shape of 40,80,1 representing width, height and channel number respectively is used. The Learning of the model is centered around three arguments, these are opti-

mization, metrics and the loss function. These functions are essential for a successful training and a better classification. The source code on Table 5.4 outlays the architectural construct of our CNNs model using the Weizmann and KTH dataset respectively. It consists of interdependent layers of convolution, activation, max-pooling, dropout and the fully connected layer which comprises of a flattened and dense output vector. However, it is important to note that while this architectural construct is peculiar to our model, experts in machine learning are responsible for determining how their CNNs architectural buildup model is designed. The choice of how much of dropout scheme needs regularization is also a design bias for machine learning engineers. Furthermore, the output shape associated with each respective layer has the tendency of shrinking down the layers. As each feature maps traverse from its input layer to the next output layer, output shape becomes smaller and this eventually equals to an output vector representing the number of classes in the recognition or classification model.

For clarity and also to understand the score metric of each model, a brief introduction of terms that are used to evaluate the results of performances is given. Below are some terms that are helpful in the definition of the score metrics:

Common Terms:

1. Positive (P): Correct prediction (for example: Running)
2. Negative (N): Incorrect prediction (for example: Not Running).
3. True Positive (TP): Actual class Observation is positive, but the predicted value is positive.
4. False Negative (FN): Actual class Observation is positive, but the predicted value is negative.
5. True Negative (TN): Actual class Observation is negative, but the predicted value is negative.
6. False Positive (FP): Actual class Observation is negative, but the predicted value is positive.

Precision: Precision is the percentage of positive correctly predicted observations to the total positive predicted observations. This is a measure of how many positive predictions were real positive observations.

$$Precision = \frac{TP}{TP+FP} = \frac{\text{positively(correctly)predicted}}{\text{Totalpositiveprediction}}$$

Recall: Recall is the percentage of appropriately predicted positive observations to all the total predicted observations.

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

F1-score: F1-score is the mean average of both the Precision and the Recall and it is particularly useful than a stand alone accuracy metric particularly when the dataset is unevenly distributed. It is denoted as :

$$F1 - Score = 2 * \frac{(Recall*Precision)}{(Recall+Precision)}$$

Table 5.9 presents a tabular detailed evaluation of our proposed loss CNN's model experimented on the Weizmann datasets, the using the Precision, Recall, and the F1 Score metrics. These score metrics are means of measuring how good our model was in relation into the classification and recognition of different class labels in our training and test data. Our goal is to empirically evaluate the functionality of our regularized model with the concatenation of dropout layers between model layers so as to further improve the power of the generalization models.

The effect of dropout on model regularization has extensively been discussed in this work and empirical analysis has been performed to further highlight how effective they can be in attaining quick convergence and a better performing model. Logging training loss and accuracy is a great way of making inferences on how well our model is converging. However, with an overfit model, the use of accuracy and loss as metrics of evaluation can also become a superficial way of evaluation. Therefore, performances on these metrics are difficult to evaluate as best accuracy on poorly regularized models can become badly scaled and this produce weak performance when tested on unseen data.

The performance metrics of both regularized and unregularized models using KTH and Weizmann dataset is discussed extensively in this section. The KTH dataset consists of six classes duly represented in Tables 5.6 and 5.7. The average score for

Table 5.4: Model Architecture for convolutional neural networks with Weizmann dataset

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='PReLU',
padding='same', inputshape=(40, 80, 1)))
convout1 = Activation('PReLU')
model.add(convout1)
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout2 = Activation('PReLU')
model.add(convout2)
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout3 = Activation('PReLU')
model.add(convout3)
model.add(MaxPooling2D(poolsize=(nbpool, nbpool)))
model.add(Dropout(0.5))
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout4 = Activation('PReLU')
model.add(convout4)
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout5 = Activation('PReLU')
model.add(convout5)
model.add(MaxPooling2D(poolsize=(nbpool, nbpool)))
model.add(Dropout(0.5))
model.add(Convolution2D(nbfilters, nbconv, nbconv))
convout6 = Activation('PReLU')
model.add(convout6)
model.add(Flatten())
model.add(Dense(128))
model.add(Activation('PReLU'))
model.add(Dense(2))
model.add(PReLU)
model.add(Activation('proposedloss'))
model.compile(loss='categorical_crossentropy',
optimizer='adadelta', metrics=["accuracy"])
```

5.6 Model Optimization

Table 5.5: A summary of model evaluation for CNNs using KTH datasets

Train on 576 samples, validate on 144 samples Epoch 1/20
576/576 [=] - 6s - loss: 1.7804 - acc: 0.2101 - valloss: 1.6016 - valacc: 0.4097
Epoch 2/20
576/576 [=] - 4s - loss: 1.4554 - acc: 0.4358 - valloss: 1.1288 - valacc: 0.5903
Epoch 3/20
576/576 [=] - 4s - loss: 1.1090 - acc: 0.6076 - valloss: 1.0872 - valacc: 0.5278
Epoch 4/20
576/576 [=] - 4s - loss: 0.8850 - acc: 0.6840 - valloss: 0.7445 - valacc: 0.7431
Epoch 5/20
576/576 [=] - 4s - loss: 0.7104 - acc: 0.7708 - valloss: 0.6306 - valacc: 0.7639
Epoch 6/20
576/576 [=] - 4s - loss: 0.5973 - acc: 0.7743 - valloss: 0.5217 - valacc: 0.7986
Epoch 7/20
576/576 [=] - 4s - loss: 0.5084 - acc: 0.8090 - valloss: 0.4502 - valacc: 0.8472
Epoch 8/20
576/576 [=] - 4s - loss: 0.4549 - acc: 0.8299 - valloss: 0.3965 - valacc: 0.8611
Epoch 9/20
576/576 [=] - 4s - loss: 0.3576 - acc: 0.8663 - valloss: 0.4501 - valacc: 0.8681
Epoch 10/20
576/576 [=] - 4s - loss: 0.3542 - acc: 0.8663 - valloss: 0.3906 - valacc: 0.8958
Epoch 11/20
576/576 [=] - 4s - loss: 0.2326 - acc: 0.9184 - valloss: 0.3991 - valacc: 0.8750
Epoch 12/20
576/576 [=] - 4s - loss: 0.2324 - acc: 0.9201 - valloss: 0.4330 - valacc: 0.8958
Epoch 13/20
576/576 [=] - 4s - loss: 0.1902 - acc: 0.9306 - valloss: 0.3544 - valacc: 0.9097
Epoch 14/20
576/576 [=] - 4s - loss: 0.1670 - acc: 0.9444 - valloss: 0.4365 - valacc: 0.9333
Epoch 15/20
576/576 [=] - 4s - loss: 0.1935 - acc: 0.9358 - valloss: 0.3601 - valacc: 0.9097
Epoch 16/20
576/576 [=] - 4s - loss: 0.1810 - acc: 0.9340 - valloss: 0.3022 - valacc: 0.9506
Epoch 17/20
576/576 [=] - 4s - loss: 0.1339 - acc: 0.9531 - valloss: 0.4177 - valacc: 0.8889
Epoch 18/20
576/576 [=] - 4s - loss: 0.1533 - acc: 0.9410 - valloss: 0.3620 - valacc: 0.9867
Epoch 19/20
576/576 [=] - 4s - loss: 0.1545 - acc: 0.9462 - valloss: 0.3464 - valacc: 0.9597
Epoch 20/20
576/576 [=] - 4s - loss: 0.1278 - acc: 0.9479 - valloss: 0.3501 - valacc: 0.9628
Test score: 0.350081033177
Test accuracy: 0.962777777778

Table 5.6: Evaluation metrics with CNNs model for KTH dataset

Activities	Precision	Recall	F1-Score	Support
class 1(WALKING)	0.92	0.92	0.92	25
class 2(JOGGING)	0.95	1.00	0.98	21
class 3(RUNNING)	0.96	0.92	0.94	27
class 4 (BOXING)	1.00	1.00	1.00	27
class 5(HAND WAVING)	1.00	0.95	0.97	20
class 6(HAND CLAPPING)	0.96	1.00	0.98	24
avg / total	0.97	0.97	0.97	144

Table 5.7: Confusion matrix of the recognition evaluation in % using convolutional neural networks for different activities of the KTH database

Activities	Walking	Jogging	Running	Boxing	Hand Clapping	Hand Waving
Walking	23	1	1	0	0	0
Jogging	0	21	0	0	0	0
Running	2	0	25	0	0	0
Boxing	0	0	0	27	0	0
Hand Clapping	0	0	0	0	19	1
Hand Waving	0	0	0	0	0	24

all three-evaluation metrics are 97%, for the precision, recall and the F1-Score respectively. The class activity of running and boxing had a precision score of 96% and 100% respectively while their recall score for jogging, boxing and hand clapping were 100% except for the hand waving with a 95% score. Interestingly, from the results analysis between the two datasets, it was observed that our CNNs model performs better with the Weizmann dataset than with the KTH dataset. A more cogent explanation for this differential are more likely to results from the number of KTH training sample that was used in training as they were fewer than the Weizmann datasets. This result underscores the empirical finding earlier mentioned in the literature review section that deep networks are better with more training data as this improves the convergence of the model and also aids better recognition and classification task.

The graphs in Figure 5.14 demonstrate that dropout is an effective regularization technique that leverages on the problematic effect of overfitting. It also allows quick convergence when used correctly. A comparative difference between models in Tables

5.1 and 5.4 is that dropout is introduced in certain hidden layers to reduce the number of parameters that are likely to co-adapt as discussed in [196, 200, 201]. Figures 5.14(a) and 5.14(c) represent accuracy and the optimization scores derived from the introduction of dropout in the model while Figures 5.14(b) and 5.14(d) describe a model that has no dropout layers between the hidden layers of the convolutional neural network.

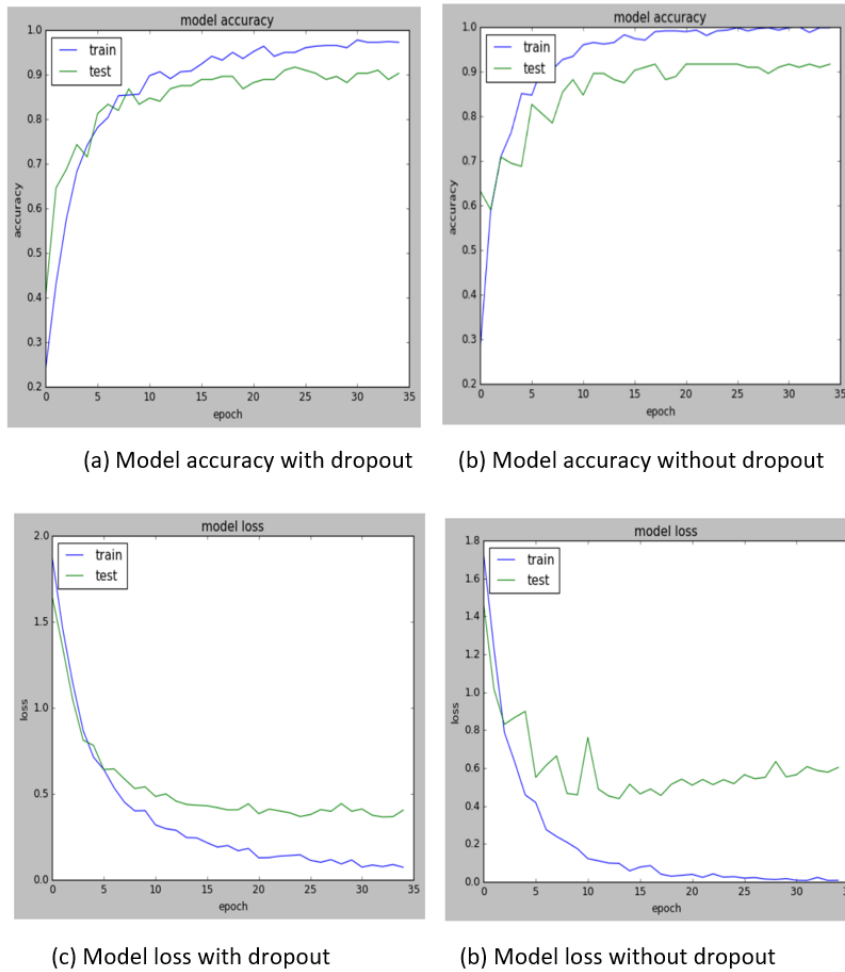


Figure 5.14: Dropout Effect on KTH - Effect of dropout on model accuracy and loss on the KTH dataset

A detailed analysis of the KTH dataset model shown in Figure 5.14 (a) and Figure 5.14(b) shows that both model accuracies are approximately the same, but a close look at Figure 5.14(a) at epoch 35 shows that the graph line has started to gain more accuracy. Moreover, the convergence rate for the regularized model of Figure 5.14 (a)

is smooth and is at close proximity with the training graph. This tells a better story that the regularized model is more inclined to generalize with unseen data than the unregularized model graph seen in Figure 5.14(b). The convergence rate seen in the unregularized model is quite unstable and is characterized by saw tooth shape. This is a clear demonstration of overfitting in Figure 5.14(b). This uneven zig-zag pattern in the unregularized deep learning model creates difficulty for quick convergence in the design of any machine learning model. All these are known pointers which measure how well the dropout technique can become a veritable method of resolving the ugly concept of overfitting issues common in modern problems of machine learning. This model type generalizes even better as seen in Figures 5.14(a) and 5.14(c) when unseen data set is used for testing and validation of the model. The optimization scores are presented in Figures 5.14(c) and 5.14(d). This represents the losses associated with each model. The training set loss for both models with dropout and without dropout are sharp and steep in their downward curve decay which indicates a progressive optimization process. This is obviously so because the humongous parameters of the model have a complete knowledge of the training dataset. Therefore, this model is characterized by relative ease of finding the appropriate weight and biases to maintain a steep downward curve. The high number of parameters present during training are key factors for such optimization gain. Sadly, this is a major problem in neural network and this leads to overfitting. The test set decay curves are however different as their optimization function curves are better with dropout as shown in Figure 5.14(c). Again, in close proximity, there are similarities to the training set losses. The optimization loss as seen in Figure 5.14(d) is irregular in shape and thus exhibits same zig-zag downward curve decay. A close observation of Figure 5.14(d) reveals a zig-zag decay at the initial stage which is then followed by a slight upward rise at the tail end of the graph. This is an indication of the poor model that is capable producing fantastic results with the training set but generalizes poorly on the test or validation data.

The Weizmann dataset result shown in Figure 5.15 expresses similar patterns of performance as those seen in the KTH dataset. It was observed that the overall performance with dropout outperformed the model without dropout. Figure 5.15(a) shows the accuracy of model with 99% in comparison to the test set which amounts to a 2% increment from the unregularized model with an accuracy rate of 97% as shown in Figure 5.15(b). Again, a contrast of the losses in the model as seen in Figures 5.15(c)

Table 5.8: A summary of model evaluation for CNNs using Weizmann datasets

Train on 728 samples, validate on 182 samples
Epoch 1/20
728/728 [=] - 5s - loss: 1.8272 - acc: 0.2637 - val loss: 1.6397 - val acc: 0.4615
Epoch 2/20 728/728 [=] - 4s - loss: 1.2921 - acc: 0.5426 - val loss: 0.7720 - val acc: 0.8242
Epoch 3/20
728/728 [=] - 4s - loss: 0.8116 - acc: 0.7253 - val loss: 0.5330 - val acc: 0.8242
Epoch 4/20
728/728 [=] - 4s - loss: 0.5282 - acc: 0.8214 - val loss: 0.3954 - val acc: 0.8791
Epoch 5/20
728/728 [=] - 4s - loss: 0.3976 - acc: 0.8750 - val loss: 0.2601 - val acc: 0.9396
Epoch 6/20
728/728 [=] - 4s - loss: 0.3189 - acc: 0.8846 - val loss: 0.1719 - val acc: 0.9505
Epoch 7/20
728/728 [=] - 4s - loss: 0.2381 - acc: 0.9258 - val loss: 0.1528 - val acc: 0.9396
Epoch 8/20
728/728 [=] - 4s - loss: 0.2015 - acc: 0.9451 - val loss: 0.1374 - val acc: 0.9615
Epoch 9/20
728/728 [=] - 4s - loss: 0.1729 - acc: 0.9437 - val loss: 0.1348 - val acc: 0.9560
Epoch 10/20
728/728 [=] - 4s - loss: 0.1459 - acc: 0.9560 - val loss: 0.1488 - val acc: 0.9451
Epoch 11/20
728/728 [=] - 4s - loss: 0.1483 - acc: 0.9602 - val loss: 0.1347 - val acc: 0.9505
Epoch 12/20
728/728 [=] - 4s - loss: 0.1138 - acc: 0.9739 - val loss: 0.0917 - val acc: 0.9725
Epoch 13/20
728/728 [=] - 4s - loss: 0.1136 - acc: 0.9684 - val loss: 0.0764 - val acc: 0.9780
Epoch 14/20
728/728 [=] - 4s - loss: 0.1051 - acc: 0.9739 - val loss: 0.1504 - val acc: 0.9396
Epoch 15/20
728/728 [=] - 4s - loss: 0.1001 - acc: 0.9725 - val loss: 0.0551 - val acc: 0.9780
Epoch 16/20
728/728 [=] - 4s - loss: 0.0841 - acc: 0.9794 - val loss: 0.0509 - val acc: 0.9835
Epoch 17/20
728/728 [=] - 4s - loss: 0.0736 - acc: 0.9808 - val loss: 0.0622 - val acc: 0.9780
Epoch 18/20
728/728 [=] - 4s - loss: 0.0629 - acc: 0.9876 - val loss: 0.1055 - val acc: 0.9670
Epoch 19/20
728/728 [=] - 4s - loss: 0.0617 - acc: 0.9835 - val loss: 0.0549 - val acc: 0.9835
Epoch 20/20
728/728 [=] - 4s - loss: 0.0505 - acc: 0.9876 - val loss: 0.0636 - val acc: 0.9780
Test score: 0.0636460948019
Test accuracy: 0.978021978022

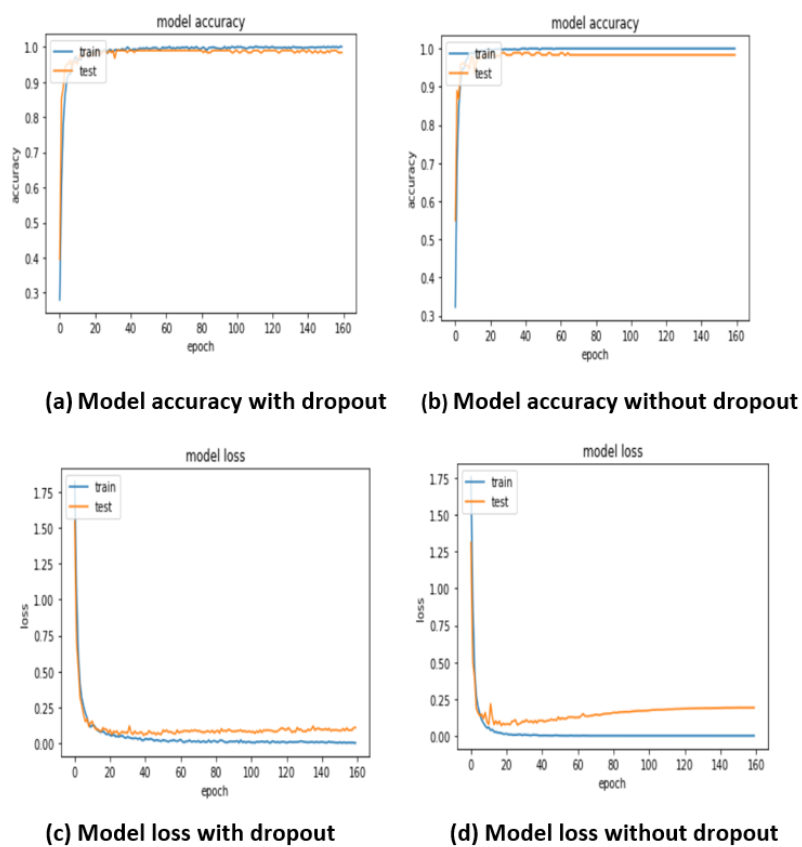


Figure 5.15: Dropout Effect on Weizman - Effect of dropout on model accuracy and loss on the Weizmann Dataset

5.6 Model Optimization

Table 5.9: Evaluation metrics with CNNs model with Weizmann dataset

Activities	Precision	Recall	F1-Score	Support
class 1(RUNNING)	1.00	0.92	0.96	25
class 2(BENDING)	1.00	1.00	1.00	24
class 3(JACKING)	1.00	1.00	1.00	34
class 4(SKIPING)	0.90	1.00	0.95	19
class 5(SIDEWALK)	1.00	1.00	1.00	28
class 6(WAVING)	1.00	1.00	1.00	23
class 7(JUMPING)	1.00	1.00	1.00	29
avg / total	0.99	0.99	0.99	182

Table 5.10: Confusion matrix of the recognition evaluation in % using convolutional neural networks for different activities of the Weizmann database

Activities	Running	Bending	Jacking	Skipping	Sidewalk	Waving	Jumping
Running	23	0	0	2	0	0	0
Bending	0	24	0	0	0	0	0
Jacking	0	0	34	0	0	0	0
Skipping	0	0	0	19	0	0	0
Sidewalk	0	0	0	0	28	0	0
Waving	0	0	0	0	0	23	0
Jumping	0	0	0	0	0	0	29

and 5.15(d) points to the fact that dropouts have the potential of reducing the effect of overfitting in deep learning architecture. To further buttress the effect of overfitting, it is observed that the decay curve for the regularized model in Figure 5.15(c) is smoother and near to the training model. The same cannot be inferred from the unregularized model in Figure 5.15(d), as the difference between the decay curve for the training and test set is larger than the regularized model. The evaluation metric in Table 5.9 has demonstrated the powerful discriminant characteristics of dropout techniques in CNNs. It has shown a near perfect classification score with an overall precision average of 99%. Furthermore, the 99% score on both Recall and F1-Score is also a clear demonstration of the quality nature of our model designed using dropout regularization on the CNNs architecture. The results from Table 5.9 and 5.10 show why the adoption and application of regularization in deep learning architecture in computer vision and image

processing sector has become popular. Its popularity stems from its manoeuvrability of hyper-parameters by regularization to achieve state-of-the-art results, in the image recognition model. All five activities of the seven classes (i.e.) bending, jacking, side-walk, waving and jumping had a perfect 100% score in all three-evaluation metrics, while the only two activities that performed slightly less than other five were the running and skipping. From the activities rows, the running column has a precision score of a 100%, while recall and F1-score were 92% and 96% respectively. The skipping activities have a similar narrative with the recall evaluation score of a 90%, while the precision and F1-score have 100% and 95% respectively. Again, a comparison between Tables 5.9 and 5.10 corroborates the result analysis discussed earlier. Therefore, going by the result analysis on the Weizmann dataset, the dropout feature regularization in deep learning algorithm in challenging an area like computer vision has exhibited its robustness and effectiveness. This experiment has gainfully elucidated the importance of dropout implementation in building deep neural network models.

5.6.3 L2 feature Regularization

Another potent and successful way of reducing overfitting is increasing the training dataset, thus reducing the network size complexity which has also shown to be very effective. However, care must be taken not to overly prune network size to stop overfitting the recognition model. Large sized networks are good for better classification purposes than smaller networks. The difficulty experienced in getting hold of larger dataset and not able to scale up our network instantaneously while training is a major concern to the leverage provided by large dataset and networks. Besides the dropout techniques of regularization, L2 technique of regularization has commonly been used in neural network optimization. This technique wraps an extra term around the cost function, causing the model weight to be penalized as they decay to get their respective optimum values. This extra term is called the regularization term, due to its affinity for weight penalization using mathematical differentiation techniques, and is often referred to as the weight decay method. This term is usually represented as shown in Equation 5.41 with the first term representing Loss in Equation 5.41 is the loss function, while the second term is the regularization term. This regularization term contains the square sum of all the weights present in the network and the summed square weight is scaled

by a factor of $\frac{\lambda}{2n}$, the λ is called the regularization parameter and must be greater than 0, n parameter in this case represents the size of the training.

$$C = Loss + \frac{\lambda \sum w^2}{2n} \quad (5.41)$$

The regularization term allows for the network to utilize small weights and large weights only if they exhibit signs of improving the cost function at the onset of each propagation. This holistic pattern seeks to define a balanced way of appropriating smaller weights and minimizing the cross-entropy function and the λ plays a significant role in the relative importance among the two elements.

The implementation of the normal gradient descent learning can be achieved in the regularized network and the computation of the partial derivatives as discussed in section 5.2.2 is like Equation 5.42. However, the addition of a regularizing term to all the partial derivatives of the weight term is what makes the difference between the backpropagation done in regularized and unregularized cost function.

$$\frac{\partial C}{\partial w} = \frac{\partial Loss}{\partial w} + \frac{\lambda w}{n} \quad (5.42)$$

$$w \rightarrow w - \eta \frac{\partial Loss}{\partial w} - \eta \frac{\lambda w}{n} \quad (5.43)$$

where η is the learning rate and λ is the regularization parameter.

$$= (1 - \frac{\eta \lambda}{n})w - \eta \frac{\partial Loss}{\partial w} \quad (5.44)$$

The expressions in Equations 5.43 and 5.44 are not different from the known gradient descent learning rule, except for the weight decay that is induced by the scaling factor $(1 - \frac{\eta \lambda}{n})$. This factor contributes towards the diffusion of its weight terms towards zero term, though the other term in the equation helps to balance the weights. The weights never reach the zero term and may even increase at one point. This whole idea in L2 regularization is the major concept of reducing unregularized cost function that will help in reducing overfitting. Improving the model generalization techniques means finding a way to improve the feature learning process. Good and stable features are important in learning internal representation in any datasets. Therefore, the L2 regularization is used to suppress overfitting which is responsible for poor generalization thus inhibiting proper representation and recognition of vital patterns which are key in HAR purposes.

5.6.4 Effect of L2 regularization on KTH dataset

This section examines the effect of L2 regularization on weight and biases of the system to determine its performance on the model. The model architecture is the same as described in Table 5.4. The model accuracy and losses are two empirical metrics that can be used to analyse how well the model is generalizing. This neural network is designed with a mini-batch size of 32, 100 hidden nodes, regularization parameter of $\lambda = 0.1$, a learning rate of 0.01 and the cost function. The cost pattern as shown in Figure 5.16 depicts the training cost and they lean towards zero mark. Meanwhile, the test data cost line initially follows similar patterns of the training set. However, there is an abrupt decline in its downward decay curve. Instead, there is progressive continuous increase in the graph line and this is an indication of model overfitting on the KTH dataset.

Similarly, a close look at the accuracy on Figure 5.17 presents a significant challenge. It can be observed that with 60 epoch, the model stops to update its accuracy. This is evident in the constant line seen on the accuracy graph in Figure 5.17. The failure of the test to achieve accuracy to continue and its accuracy update can be attributed to two main reasons. First, the model is susceptible to overfitting and secondly the neurons in the model have failed to learn new things as a result of the gradient saturation. The only inference that can be learnt from the latter is that unregularized model can cause neurons to be stuck in the local minimum. This can prompt dead neurons, causing gradient update to be impossible.

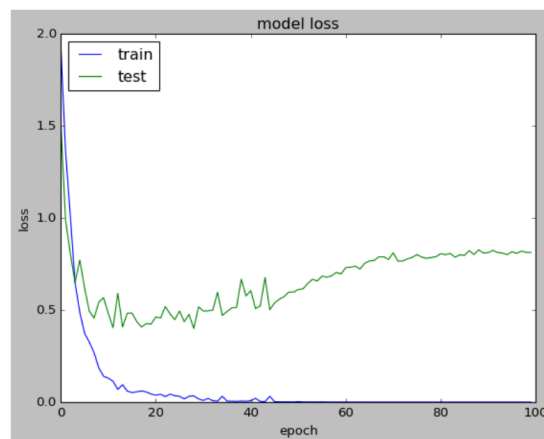


Figure 5.16: Loss function Graph - Unregularized Cost

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

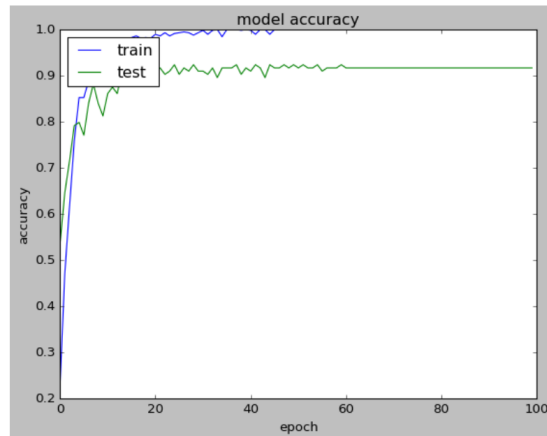


Figure 5.17: Accuracy Graph - Unregularized Accuracy

In Figure 5.18, the application of regularization helps to conquer the disadvantages caused by overfitting. The cost of both the training and the test set has similar decay patterns and this is an indication that the model has suppressed overfitting. Furthermore, there are provisions and rooms for improving the accuracy. The accuracy in Figure 5.19 shows a continuous tendency of improving its accuracy as opposed to the constant horizontal line seen in Figure 5.17. This implies that with adequate knowledge of how to choose the correct learning rate and other hyper-parameters, a better accuracy is guaranteed. However, though the accuracy of both the unregularized and the regularized are almost the same at 91%, the gain of the regularized results is more achievable because they can generalize well with unseen data considering the performances as seen from the regularized graphs.

In conclusion, with empirical analysis highlighted in this work, it can be said that regularization is a reasonable cause that has allowed our network to generalize well over unregularized network.

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

Parameter optimization is another reliable way of contributing to the successful application of the deep learning process. To achieve the most optimum results, efforts

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

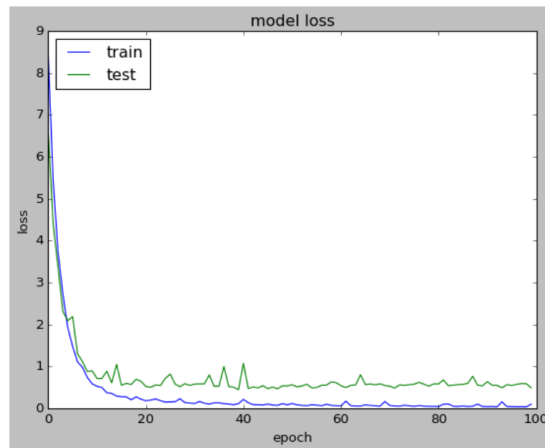


Figure 5.18: Regularized Cost - Graph Showing a Regularized Loss

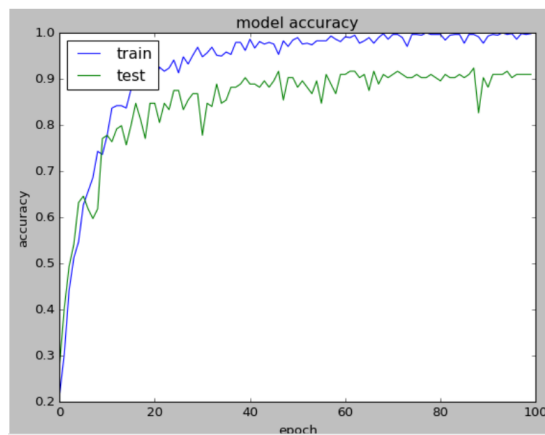


Figure 5.19: Regularized Accuracy - Graph Showing a Regularized Accuracy

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

at choosing appropriate parameters are often seen as key for creating better learning models. Each of these parameters is often called hyperparameter. Examples of these hyperparameters can be number of layers present in the network, number of neurons in each layer, activation function type, learning rate and number of call or epoch. All these hyper parameters have unique functionality that can steer and influence the overall performance of any model. An optimum hyperparameter choice in machine learning function can decide what machine algorithm should be called state-of-the art. Hyperparameters functional capabilities are unique, and they vary from one dataset to another dataset. There are several methods by which machine learning model can be tuned via the use of hyperparameters.

Hand tuning and grid search methods are some of the earliest methods of performing hyper parameter optimization[185]. Hand tuning is characterized by a set combination of parameters chosen to influence the optimum path to a better recognition and classification result and these performance metrics are recorded. This hand tuning method is continuously performed with the intention of discovering optimum parameters capable of improving the overall classification or eventual recognition of the results. The chance of obtaining better results depends on the robust acquaintance of the knowledge domain with the subject area and sometimes by mere probability of chance and time. These kinds of parameters searching are very time-consuming and difficult to realize. The grid search method is a better refined way of searching good hyperparameters. This method allows for each participating parameter to be divided into an evenly spaced range. Every unit and all possible combinations within the parameter range are explored for identifying the most significant parameters that allow high representational learning from any data possible. The computational burden of this type of method is performed by computers. However, such computation becomes time consuming as any slight increment in the parameters-combination can exponentially grow the computational process. For example, tuning 5 hyperparameters in a model with a possible combination of 10 values will amount to 10^5 parameter-combinations. With the addition of only one more hyperparameter, there will be 10^6 parameter-combinations. Another type of optimization scheme in deep learning is called random search method which is, unlike the grid search. This method randomly parses the hyperparameter space to locate the best hyper-parameter combination. The probability of targeting

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

the optimum hyper-parameters decreases to zero as the number of combinations increases leading to poor generalization of the model. The particle swarm optimization uses the population-based heuristic algorithm that mimics the collective behaviour of groups of entities like animals. Examples are the school of fish, colony of ants, and flocks of bird and bees swarming. All these are directional movements to a defined position with the objective of achieving a common goal in a multidimensional space. In [202], this method was found to be simple. It also retains the capability of locating optimum hyperparameters in multidimensional spaces.

The Gaussian process is a distribution function often used by the Bayesian optimization algorithm. It has been used successfully to improve the learning rate in the deep architectural learning model. In this section, an attempt to implement and empirically evaluate the Bayesian process in our model is discussed. Bayesian optimization can be said to belong to a group of optimization algorithm known as the sequential model-based optimization algorithm. This algorithm is built around the utilization of previously known observation of the loss function f to decide the next optimal region that could be better in optimizing loss function f .

There are two vital choices that must be made when Bayesian optimization is considered for the optimization of the function. Firstly, the posterior expectation of the function f can be calculated from previously evaluated parameters on given points. This posterior expectation can be modeled by the Gaussian process prior, known for its flexibility as it can easily be managed. Secondly, an acquisition function is constructed for probing into new points for the loss function to maximize certain utilities of the expectation of f . The utility parameters are good pointers to the next best domains that are highly probable for the optimum sampling of the loss function f . A continuous repetition of these steps is allowed pending when some convergence measure is actualized.

5.7.1 Learning Process

In this section, our goal is to experiment and improve the feature learning using the Gaussian process for effective parameter selection in HAR. Human activity recognition can become very intractable due to the complexity of body pose variation and high correlation of pixel values. A slight change in body pose can present a significant shift in model inference. Deep learning to some extent has handled such complexity due to it

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

high representational mapping power. However a lot is still needed to improve the model accuracy and convergence rate for inference model. Therefore, exploring additional ways of maximizing the best practice of automatic hyperparameter tuning is invaluable to this research work. Hyperparameter tuning has been used in many areas, including computer vision, natural language processing and other areas of artificial intelligence for obtaining state-of-the-art results. In view of its wide range of applications, we aim to direct this technique to experimentally seek more ways of tuning the model for better function approximation.

In this work, a comprehensive evaluation of the hyperparameter space search with Bayesian process is undertaken to enable this study add knowledge to the domain of the computer vision. This research limits its hyperparameters space search to the learning-rate of the optimizer, number of dense layers, activation function choice and the number of nodes present in each dense layer. The scikit-optimize application program interface (API) from the python package is used in the automation of the parameter space and aim to automate the whole process given the resources of computation that are currently available. The first stage is to define a valid search dimension for all these hyperparameters enumerated earlier. The learning rate search space is created using logarithmic transformed values between the lower and upper bound search range chosen for the optimization process. For this particular experiment and for the purpose of simplicity, we define the learning rate search range between $1e^{-2}$ being the upper bound value and $1e^{-6}$ as the lower bound value. Another important hyperparameter is the number of fully connected (dense) layers that are present in the design of deep learning architecture as discussed in section 5.2.8. The effect of varying the number of dense layer is monitored. As it relates to the dense layer, the search dimension is between 1 for the least densed model and 5 for the most densed layers. The number of nodes present in each fully connected (dense) node is another factor that probably influences the generalization ability of the model. Therefore, a search-dimension for the number of nodes is allowed to range from between 5 and 512 nodes respectively. The activation choice for the hyperparametrization is between the Tanh, Relu and Sigmoid function. Table 5.11 shows a tabular representation of the hyperparameters used in the optimization of the HAR model.

A default search-space parameter is shown in Table 5.12 used to initialize the optimization start point. These choice parameters are determined either by individuals

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

Table 5.11: Hyper-parameters summary and dimensional search range

Hyper-Parameters	Search-Dimension
Learning Rate	Range($low = 1e^{-2}, High = 1e^{-6}$)
Number of Dense Layer	low =1, High= 5
Number of Nodes in each Layer	low =1, High= 5
Activation function type	Categorical = ' <i>Relu</i> ' ' <i>Sigmoid</i> ' ' <i>Tanh</i> '

Table 5.12: Stochastic hyper-parameter selection for initial training

Hyper-Parameters	Search-Dimension
Learning Rate	$1e^{-6}$)
Number of Dense Layer	1
Number of Nodes in each Layer	16
Activation function type	<i>Relu</i>

hand tuning experience or by heuristic method. However, it is better to get the initialization search space right to enable a quick convergence and to achieve better generalization of the model. The model is trained and its fitness function evaluated on the test set of KTH human activity database. This architectural model is like the earlier one proposed except for the new parameters function introduced. At the end of the training and the fitness process, the best parameter is saved and only replaced when there is a better accuracy metric from other subsequent iterations.

With the hyperparameter dimensional search defined, the Bayesian optimization techniques is used to further implement the optimization process. The Gaussian process fed to the expected improvement as its acquisition function is used to model the probability distribution over the loss function derived from our posterior. For the sake of simplicity, this work sets the number of calls to 50 at the beginning. Increasing the number of calls has also shown better results which is a clear indication that the likelihood of obtaining optimal accuracy can directly be linked to the average number of time spent in hyperparameter space exploration. The result displayed in Table 5.13 shows the accuracy of the model only when the default parameters in Table 5.12 which were obtained with heuristic stochastic selection are used. With a fewer epoch, an accuracy of 35.62% was attained with the use default parameters.

However, this accuracy obtained is just a reference point which later forms the basis and informs the direction from which the Gaussian process chooses its next hyperparameters values from. A full hyperparameter dimensional search space defined earlier is then used to fit the convolutional neural network model. With this process, the hyperparameters search space was able to produce better results compared to earlier accuracy gain

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

Table 5.13: A summary of CNNs evaluation using stochastic default hyper-parameter with KTH datasets

```
learning rate: 1.0e-05
num dense layers: 1
num dense nodes: 16
activation: relu
Train on 509 samples, validate on 219 samples Epoch 1/12
509/509 [=] - 2s 3ms/step - loss: 13.5005 - acc: 0.1572 - valloss: 13.5869 - valacc: 0.1187
Epoch 2/12
509/509 [=] - 2s 3ms/step - loss: 12.5072 - acc: 0.1100 - valloss: 12.7867 - valacc: 0.0959
Epoch 3/12
509/509 [=] - 2s 3ms/step - loss: 11.8387 - acc: 0.1650 - valloss: 12.3002 - valacc: 0.1598
Epoch 4/12
509/509 [=] - 2s 3ms/step - loss: 11.4356 - acc: 0.2200 - valloss: 11.6849 - valacc: 0.2146
Epoch 5/12
509/509 [=] - 2s 3ms/step - loss: 10.8416 - acc: 0.2692 - valloss: 10.7119 - valacc: 0.2329
Epoch 6/12
509/509 [=] - 2s 3ms/step - loss: 9.2407 - acc: 0.2731 - valloss: 7.9449 - valacc: 0.2466
Epoch 7/12
509/509 [=] - 2s 3ms/step - loss: 7.4150 - acc: 0.2240 - valloss: 6.0555 - valacc: 0.2968
Epoch 8/12
509/509 [=] - 2s 3ms/step - loss: 6.2309 - acc: 0.2908 - valloss: 6.0950 - valacc: 0.3105
Epoch 9/12
509/509 [=] - 2s 3ms/step - loss: 5.4971 - acc: 0.2888 - valloss: 5.3355 - valacc: 0.2922
Epoch 10/12
509/509 [=] - 2s 3ms/step - loss: 5.1408 - acc: 0.3281 - valloss: 4.9531 - valacc: 0.3333
Epoch 11/12
509/509 [=] - 2s 3ms/step - loss: 4.4754 - acc: 0.3399 - valloss: 4.0404 - valacc: 0.3607
Epoch 12/12
509/509 [=] - 2s 3ms/step - loss: 3.9372 - acc: 0.3477 - valloss: 3.4463 - valacc: 0.3562
Accuracy: 35.62%
-0.35616438288122554
```

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

using only one hyperparameter designed by hand tuning. Table 5.14 shows a combination of different hyper-parameters using the Bayesian optimization process, the highest accuracy of 95.890% were obtained by two different hyperparameters set. These two hyperparameters, indicating learning rate, no.of dense layer, no.of nodes, activation function respectively, $(-0.95890411231071437, [8.9022070204672089e-05, 4, 193, 'relu'])$ and $(-0.95890411231071437, [0.0012575645304840357, 4, 111, 'relu'])$ gave the best optimum accuracy. However, the latter of these two was the most efficient as this hyperparameters type uses 111 nodes and a small learning rate compared to the other with 193 nodes and a higher learning rate. Again, the one with fewer nodes almost certainly guarantees better generalization as fewer nodes discourages model co-adaptation known for overfitting. Therefore, from this result, one can infer that Relu activation function proves to be a better activation function than the Sigmoid function using convolutional neural network. Figure 5.20 represents a pictorial representation of the convergence plot which is a clear illustration of how fast Bayesian optimization can quickly converge toward the optimum parameters that are able to produce the best accuracy in the process of recognition in HAR.

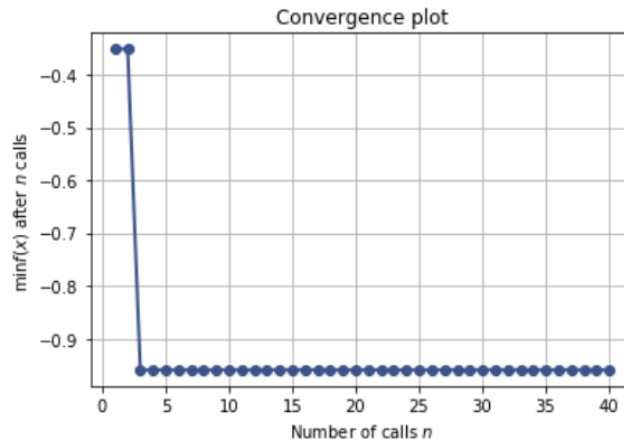


Figure 5.20: Convergence Graph - Convergence plot on the number of calls

In this research work, a careful examination and how significant the number of epochs or calls are considered. The convergence rate from this analysis in Figures 5.21 and 5.22 underscore the fact that each process of searching for optimum hyperparameters is purely stochastic. From Figure 5.21, 96% objective function minimization was reached in just about 3-4 calls. The 96% minimization rate is maintained till there

5.7 Hyperparameter Learning for improving Recognition in Human Activity Recognition using Bayesian Optimization

Table 5.14: Results associated with running hyper-parameters optimization on CNNs model for human activity recognition

[(Accuracy Metrics, [Learning Rate, No.of Dense Layer, No.of Nodes, Activation Function])]				
(-0.95890411231071437, [8.9022070204672089e-05, 4, 193, 'relu'])				
(-0.95890411231071437, [0.0012575645304840357, 4, 111, 'relu'])				
(-0.94977169221939017, [0.00024248365800359827, 5, 194, 'relu'])				
(-0.93607306017723257, [5.7836284006087636e-05, 5, 181, 'relu'])				
(-0.93607305745555935, [1.92238997301301e-05, 5, 457, 'relu'])				
(-0.93607305745555935, [2.0106501827203483e-05, 5, 488, 'relu'])				
(-0.93607305745555935, [9.2135394355971785e-05, 5, 195, 'relu'])				
(-0.93150685013157053, [8.2640806929749455e-05, 5, 255, 'relu'])				
(-0.9315068474098972, [6.1273623990882383e-05, 5, 181, 'relu'])				
(-0.9315068474098972, [6.8059825514600044e-05, 5, 182, 'relu'])				
(-0.92694063736423515, [0.00014891584897492552, 4, 165, 'relu'])				
(-0.922374427318573, [5.9477388386674411e-05, 5, 179, 'relu'])				
(-0.91780821727291084, [1.9674936809902542e-05, 5, 489, 'relu'])				
(-0.91324201267059535, [7.3302740576439404e-05, 5, 181, 'relu'])				
(-0.91324200994892213, [2.9348165378917484e-05, 5, 199, 'relu'])				
(-0.90410958985759793, [7.7792540942874875e-05, 5, 113, 'relu'])				
(-0.89497716976627373, [6.3491856120987991e-05, 4, 181, 'relu'])				
(-0.69863013480896274, [0.0004294715885048197, 1, 158, 'sigmoid'])				
(-0.66666666721100132, [5.3004032251317876e-06, 3, 244, 'relu'])				
(-0.44748858488313686, [1.0530282317634178e-06, 5, 511, 'relu'])				
(-0.35159817474073474, [1e-05, 1, 16, 'relu']),				
(-0.35159817474073474, [1.8136240256027347e-05, 1, 496, 'sigmoid'])				
(-0.31050228432977578, [1.5778184556290273e-06, 4, 342, 'relu'])				
(-0.23287671253289263, [8.1292226070856812e-05, 5, 29, 'relu'])				
(-0.2146118714657004, [1.0709919551519687e-06, 1, 193, 'relu'])				
(-0.17351598098669965, [0.0002608178467253554, 2, 472, 'relu'])				
(-0.1552511417566369, [0.0048516890520873164, 3, 70, 'relu'])				
(-0.14155251148356696, [5.529764193402135e-05, 4, 180, 'relu'])				
(-0.14155251148356696, [0.0087902516010522885, 1, 502, 'relu'])				
(-0.14155251066706495, [9.5841536576673968e-05, 2, 197, 'relu'])				
(-0.14155251066706495, [0.0031892600523502367, 4, 140, 'sigmoid'])				
(-0.14155251066706495, [0.0098150172443698808, 1, 393, 'sigmoid'])				
(-0.13698630150594668, [3.4407931419560163e-05, 3, 374, 'sigmoid'])				
(-0.13698630150594668, [7.8957956748252939e-05, 4, 481, 'sigmoid'])				
(-0.13698630150594668, [0.00010795946087768507, 5, 226, 'sigmoid'])				
(-0.13698630150594668, [0.0014576917030894978, 4, 110, 'relu'])				
(-0.13242009146028458, [0.0081495512628050239, 5, 195, 'relu'])				
(-0.11872146122123553, [0.0001011775331869387, 2, 124, 'sigmoid'])				
(-0.11872146122123553, [0.001413508723813464, 2, 59, 'sigmoid'])				
(-0.11872146122123553, [0.0073208191085134371, 5, 26, 'relu'])				

were about 16 epochs and there was no significant convergence towards optimum hyperparameter value. Furthermore, a slight minimization of the objective function is achieved between the 16 and 30 epochs thus pushing the objective function towards the 97% mark and 98% respectively. Therefore, this model achieved convergence of the optimum values just within 31 epochs. This is proof that the Gaussian process is valuable in the optimization processes of the deep learning architectural model. However, a call of 200 epochs has shown no superior minimization of its objective function over that of Figures 5.21 and 5.22 presented 4 visible stages of convergence to optimum hyper-parameters. Though the same minimization value of the objective function was reached like that of Figure 5.21, it can be said that choosing the number of epochs to run can become another hyperparameter that needs careful selection. This is meant to avoid the unnecessary burden of computation that may be passed to the model. Therefore, while it can be said that increasing the number of epochs is good for best hyperparameter location, a considerable number of epochs can also achieve optimum convergence.

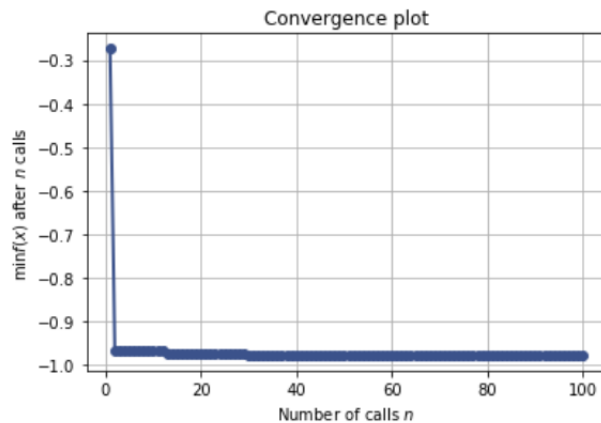


Figure 5.21: Convergence Graph for 100 - Convergence plot with 100 epoch

5.8 Summary and Discussion

In this chapter, we proposed a new regularization term for the center loss function to improve class discrimination in HAR. This new regularization term is developed by assuming the extra log-prior-distribution as a regularizer and is known for updating prior

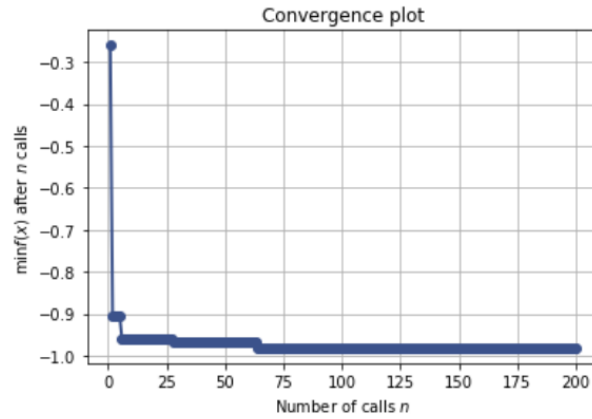


Figure 5.22: Convergence Graph for 200 - Convergence plot with 200 epoch

knowledge in parameter estimation in both Gaussian Mixture(GM) and Maximum-a-Posteriori Estimation(MAP). By combining the extra log-prior-distribution as a regularization term with the supervisory loss function as witnessed in both Weizmann and KTH has made a tremendous improvement in the discriminative power of deeply learned features in HAR.

The work done in this chapter has shown the numerous advantageous opportunities that are available when deep learning algorithms are used in computer vision. The automation and other numerous hyper-parameter tuning varieties presented by deep learning made it a unique technique which researchers are actively exploring and as such will remain dominant in artificial intelligence. The projection of such continuous dominance has necessitated this research work to look at a holistic view on the best practices in this subfield. The move towards highlighting algorithms to improve a deep learning model is considered not only important but also as a strategic pathway that will further advance the deep learning frontiers. Therefore, this chapter has carefully elucidated the fundamental composition of deep learning makeup, its building blocks and how all the components are all inter woven together to build a successful recognition model. This chapter has also presented tactical and skillful methods of mining representation features that are pivotal in obtaining state-of-the-art recognition model. The regularization of these learned features, its gains, and improvements as presented in this chapter is a clear demonstration that underscores the importance of feature regularization in the deep learning model.

5.8 Summary and Discussion

An improved best hyper-parameter blend and accurate regularization technique can become a remedy for finding a state-of-art recognition model for HAR. Although an adequate attempt has been made to explicitly explain all the reported research discoveries, we cannot emphatically suggest that these results should be chiselled in stones as different model architecture and datasets can produce results that could vary from the one reported in this thesis.

6

Conclusion and Future works

6.1 Summary of work

In this section, the progress of research, contribution to knowledge and the direction of future studies have been succinctly highlighted. There is a quest and race by most notable organization such as Google, Amazon, Facebook and other upcoming artificial intelligence organizations to remain a dominant force in this area is tough. However, it has become a critical stage for the next technological frontiers . Such new frontiers have given birth to some of the recent technological breakthroughs like self driving cars, various human activity recognition (HAR) machines, autonomous machines, natural language processing and other advances in the computer vision sector. To attain speedy technological development in these areas, more research needs to be framed into this concept of artificial intelligence, as more growth has been projected to occur in this field as time roll by. Current global safety challenges in different geographical regions have allowed enormous investments to be directed towards the study of criminal tendency. The quest for the recognition of human activity using technology can reasonably be projected to have valuable utility in public places like bus stations, patient monitoring to more complex scenes such as crowd monitoring and border security. This research has dedicated its study to various methods of feature learning and regularization toward promoting adequate recognition and optimization of the process of human activities in the field of computer vision. This study has only limited its findings to the recognition of human actions. Hence the dataset obtained from the KTH and Weizmann dataset were used as our datasets benchmark.

The dynamics of criminality, terrorism and other such negative vices that are inimical to society are tremendously evolving. All the chapters of this thesis have significantly attempted to proffer the state-of-the art solution with the use of cutting edge advances in feature regularization. It is paramount to note that while urban surveillance phrase in the studies was a delineation term that generally refers to a subset of monitoring human activity, all the models and technology represented in this work are valuable in the other sectors of computer vision and image processing.

Timely and robust detection and classification of human action and their activities can become a clear distinction between life and death, both in crime prevention or in other sectors where monitoring of human activities is inevitably important. Hence, chapter two of the thesis was dedicated to the purpose of explicitly understudying various research incursions in the recognition of human activity. An in-dept analysis done in the literature review has revealed that most studies on regularization were done on the within-class matrix. It was recorded in literature that the within-class matrix constitutes the prime problem for poor performance in recognition. The rapid decay nature of the eigenspectrum due to the small sample size problem, presence of noise in the datasets, occlusion, high dimensionality and correlation in image pixels were found to be the major reasons why the within-class matrix in sympathy to these constraints suffers from poor performance. Considering these catalogued problems in the within-class matrix, the analysis in chapter four proffers crucial solutions to these problematic issues and the results obtained show that the excellent recognition model can be built with the right parameter selection for the regularization term.

Again, the extensive literature review done was instrumental to the exploration and adventure into researching the deep learning methodology, feature learning and regularization in the computer vision sector. This knowledge and understanding derived from literature helped in the redirection of this research strength towards current trends and state-of-the art methodology of using the deep learning concept to solve complex and difficult HAR problems as was done in chapter five. This research study has also breached and improved statistically the vast pool of Computer Vision domain knowledge, thus beaming more energy towards the course of answering and narrowing the research question "how effective were hand crafted features in performing HAR, as compared to an end-to-end hierarchical deep learning neural networks". A comparison between the results obtained in Chapter four and five evidently point to the fact that

the deep learning feature methods like the CNNs are by far more easier to develop because they do not automatically extract learnable features, but have a large range of hyper-parameters that can be varied to suit the desired outcome. Additionally, the deep learning method takes lesser time in its architectural frame up than the hand crafted method as described in chapter four. While feature learning and regularization in the deep learning method encompasses a high degree of parameter freedom, the same cannot be said for the hand-crafted model. Hand-crafted feature parameters are often fixed and very difficult to be crafted as they have low parameter freedom, thus making it more difficult to reach optimum performance of the recognition model. However, this research has demonstrated that detailed planning and adequate feature engineering knowledge process, hand-crafted features will perform well in human activity recognition just as its deep learning counter-part with the two datasets. Human activity class discrimination is highly dependent on the quality of the picture pixels used for training in our model, hence image processing is also a huge part of this research work.

Chapter three discusses the detection of image and preprocessing of video frames to ensure better image quality and classification tasks. The task of building a classification and a recognition model must begin with obtaining the object of interest. This task has been made achievable by the simple method of background subtraction. Though another notable option for obtaining the object of interest discussed in this thesis was the use of Gaussian mixture model, the latter of the two was concentrated on as the videos contain less occlusion and background variation. The image cleaning improved the results obtained and focusing only on the object of interest has also helped to circumvent the tedious computational process that would have been necessary had all the pixel values been present in the video frames which were used in the classification of the human activities. Though the deep learning processing has more capacity and robustness of computation, the same cannot be said for the shallow and hand crafted models whose algorithms are limited in terms of computational power. This research has opened up avenues for demonstrating that intelligence and automation in the recognition system of cameras can be improved. This concept was actively and greatly researched in chapter four and five, both of which show that extracting quality features and using them for the purpose of discriminating different human action is a key success factor in computer vision. The eigenspectrum regularization of the within-class matrix with the proposed four-parameter regularization method was key

in the gains of this model recognition rate. The use of four parameter modelling scheme has allowed for a robust, reliable and precise eigenvalue prediction. Other regularization methods lack this innovative regularization scheme whereas most regularization methods carried out subspace fragmentation before regularization was done separately and in a piecewise manner. This thesis has produced reliable design and a more accurate model that reflects the true variances in the within-class matrix. Such modelling of the true variance helps avoid piecewise fragmentations that are tedious and open to errors in the eigenspectrum estimation. Therefore, the difficulty of subspace piecewise annotation before regularization has been avoided and eliminated. The technique has enabled the within-class matrix often seen as the key culprit for poor recognition purposes in HAR, to be more discriminative and active. The research has developed and demonstrated the craftsmanship to outwit some of its very problematic navigations notoriously known for challenging better classification and recognition of HAR. The empirical evidence recorded from the eigenfeature regularization method shows its effectiveness over other subspace methods like PCA, FLDA, and the recent state-of-the-art three-parameter method used for HAR. In chapter five, convolutional neural network which proved to be a power house in the feature representation and learning was duly represented in this study. Various regularization methods were implemented to reduce the common problem of overfitting which is a well known obstacle in model generalization. Experimental analysis has demonstrated that deep learning is great in model accuracy, although such accuracy may not be useful in the absence of well tuned and regularized features. This research has highlighted the trade-off which exist between accuracy and generalization and the conclusion is that care must be taken to ensure that there is a balance between these two performance metrics in order to have a robust and stable deep learning network.

The progress made by convolutional deep learning of human action recognition as seen in chapter five acknowledges the strength of the regularization of deep learning network. To this effect, It displays a clear exhibition of its dominance and is mostly preferred as technology of choice when it comes to the creation of smart and intelligence machines. These technologies are growing fast and becoming part of everyday life as computer vision technologies have found useful practices in activity recognition in big cities, robotics navigation in the industries and in automobile autonomous movements. These technologies will be around for many years to come and will require better

research innovations, funding and development to get to their next level of development. Indeed, humans are making good their promises to make machines see, understand their environment and provide excellent decision.

6.2 Future works

In the future, this method of feature learning and regularization will be extended to 3-D real-world data images which is fast becoming an area of interest for image processing researchers. Secondly, this method has delineated only single action recognition, thus providing a baseline for other complex actions to be investigated (two person interaction or group activity recognition). Therefore, future work needs to include the introduction of our four-parameter eigenvalues regularization method for the purpose of the classification of complex actions in familiar scenes of human activities. Finally, feature hybridization seeks to compensate for the shortfalls in different methodologies.

References

- [1] MEGHA AGARWAL AND PETER FLACH. **Activity recognition using conditional random field.** In *Proceedings of the 2nd international Workshop on Sensor-based Activity Recognition and Interaction*, page 4. ACM, 2015. 2, 30
- [2] JAKE K AGGARWAL AND QUIN CAI. **Human motion analysis: A review.** In *Nonrigid and Articulated Motion Workshop, 1997. Proceedings., IEEE*, pages 90–102. IEEE, 1997. 1, 4, 13
- [3] YOSHUA BENGIO, AARON COURVILLE, AND PASCAL VINCENT. **Representation learning: A review and new perspectives.** *IEEE transactions on pattern analysis and machine intelligence*, **35**(8):1798–1828, 2013. 1, 28
- [4] WILLIAM BRENDEL AND SINISA TODOROVIC. **Learning spatiotemporal graphs of human activities.** In *2011 International Conference on Computer Vision*, pages 778–785. IEEE, 2011. 2, 29, 98
- [5] BAPPADITYA MANDAL AND HOW-LUNG ENG. **Regularized discriminant analysis for holistic human activity recognition.** *IEEE Intelligent Systems*, **27**(1):0021–31, 2012. 2, 3, 12, 16, 18, 19, 48, 57, 59, 62, 63, 64, 66, 71
- [6] CHRISTIAN MICHELONI, PAOLO REMAGNINO, HOW-LUNG ENG, AND JASON GENG. **Intelligent monitoring of complex environments.** *IEEE Intelligent Systems*, **25**(3):12–14, 2010. 64
- [7] ZAFAR ALI KHAN AND WON SOHN. **Feature extraction and dimensions reduction using R transform and principal component analysis for abnormal human activity recognition.** In *Advanced Information Management*

-
- and Service (IMS), 2010 6th International Conference on*, pages 253–258. IEEE, 2010. 14, 16, 57, 59, 62
- [8] DAIRAZALIA SÁNCHEZ, MONICA TENTORI, AND JESÚS FAVELA. **Activity recognition for the smart hospital.** *IEEE intelligent systems*, **23**(2):50–57, 2008.
- [9] AHMAD JALAL AND SHAHARYAR KAMAL. **Real-time life logging via a depth silhouette-based human activity recognition system for smart home services.** In *Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on*, pages 74–80. IEEE, 2014. 2, 12, 57
- [10] MUHAMMAD HASSAN, TASWEER AHMAD, AND SADAF ALI. **Comparative analysis study of human activity recognition using various techniques.** In *Multi-Topic Conference (INMIC), 2014 IEEE 17th International*, pages 83–86. IEEE, 2014. 3, 6, 16, 36
- [11] ALESSIO DORE, CARLO S REGAZZONI, ET AL. **Interaction Analysis with a 1 Bayesian Trajectory Model.** 2010.
- [12] HOSSEIN HAJIMIRSADEGHI, WANG YAN, ARASH VAHDAT, AND GREG MORI. **Visual recognition by counting instances: A multi-instance cardinality potential kernel.** In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2596–2605, 2015.
- [13] CEM DIREKOLU AND NOEL E OCONNOR. **Team activity recognition in sports.** In *European Conference on Computer Vision*, pages 69–83. Springer, 2012. 3, 36
- [14] MITCHELL GRAY. **Urban Surveillance and Panopticism: will we recognize the facial recognition society?** *Surveillance & Society*, **1**(3):314–330, 2002. 3
- [15] IOANNIS PAVLIDIS, VASSILIOS MORELLAS, PANAGIOTIS TSIAMYRTZIS, AND STEVE HARP. **Urban surveillance systems: from the laboratory to the commercial world.** *Proceedings of the IEEE*, **89**(10):1478–1497, 2001. 3

-
- [16] ILYA SUTSKEVER. *Training recurrent neural networks*. PhD thesis, University of Toronto, 2013. 5, 21, 33
- [17] JAMES MARTENS. **Deep learning via Hessian-free optimization**. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 735–742, 2010. 33
- [18] JAMES MARTENS AND ILYA SUTSKEVER. **Learning recurrent neural networks with hessian-free optimization**. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1033–1040, 2011. 33
- [19] JEFFREY DEAN, GREG CORRADO, RAJAT MONGA, KAI CHEN, MATTHIEU DEVIN, MARK MAO, ANDREW SENIOR, PAUL TUCKER, KE YANG, QUOC V LE, ET AL. **Large scale distributed deep networks**. In *Advances in neural information processing systems*, pages 1223–1231, 2012. 30
- [20] JIQUAN NGIAM, ADAM COATES, AHBK LAHIRI, BOBBY PROCHNOW, QUOC V LE, AND ANDREW Y NG. **On optimization methods for deep learning**. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 265–272, 2011. 5, 21
- [21] BERNHARD SCHOLKOPFT AND KLAUS-ROBERT MULLERT. **Fisher discriminant analysis with kernels**. *Neural networks for signal processing IX*, **1**(1):1, 1999. 6, 16
- [22] L LEE, R ROMANO, AND G STEIN. **Introduction to the special section on video surveillance**. *IEEE Transactions on pattern analysis and machine intelligence*, **8**:740–745, 2000. 6
- [23] DARIU M GAVRILA. **The visual analysis of human movement: A survey**. *Computer vision and image understanding*, **73**(1):82–98, 1999. 13
- [24] LUN ZHANG, STAN Z LI, XIAOTONG YUAN, AND SHIMING XIANG. **Real-time object classification in video surveillance based on appearance learning**. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 13

-
- [25] RUDRA N HOTA, VIJENDRAN VENKOPARAO, AND ANUPAMA RAJAGOPAL. **Shape based object classification for automated video surveillance with feature selection.** In *Information Technology,(ICIT 2007). 10th International Conference on*, pages 97–99. IEEE, 2007. 13
- [26] YANIV GURWICZ, RAANAN YEHEZKEL, AND BOAZ LACHOVER. **Multiclass object classification for real-time video surveillance systems.** *Pattern Recognition Letters*, **32**(6):805–815, 2011. 13
- [27] ZAFAR A KHAN AND WON SOHN. **Abnormal human activity recognition system based on R-transform and kernel discriminant technique for elderly home care.** *IEEE Transactions on Consumer Electronics*, **57**(4):1843–1850, 2011. 14, 15, 48, 59, 62
- [28] BELKACEM FERGANI ET AL. **Evaluating a new classification method using PCA to human activity recognition.** In *Computer Medical Applications (ICCA), 2013 International Conference on*, pages 1–4. IEEE, 2013.
- [29] PETER N. BELHUMEUR, JOÃO P HESPANHA, AND DAVID J. KRIEGMAN. **Eigenfaces vs. fisherfaces: Recognition using class specific linear projection.** *IEEE Transactions on pattern analysis and machine intelligence*, **19**(7):711–720, 1997. 16, 18, 58, 63, 97
- [30] JANSON HENDRYLI AND MOHAMAD IVAN FANANY. **Classifying abnormal activities in exam using multi-class Markov chain LDA based on MODEC features.** In *Information and Communication Technology (ICoICT), 2016 4th International Conference on*, pages 1–6. IEEE, 2016. 14, 59
- [31] XINGHUA SUN, MINGYU CHEN, AND ALEXANDER HAUPTMANN. **Action recognition via local descriptors and holistic features.** In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 58–65. IEEE, 2009. 14
- [32] WEI SONG, NING-NING LIU, GUOSHENG YANG, FU-HONG LIN, AND PEI YANG. **Multi-feature fusion based human action recognition algorithm.** 2015. 14

-
- [33] HONG HAN, HONGLEI ZHANG, JIANYIN GU, AND FUQIANG XIE. **Action recognition based on hybrid features**. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 621–626. IEEE, 2012. 14
- [34] DINESH K VISHWAKARMA AND KULDEEP SINGH. **Human Activity Recognition based on Spatial Distribution of Gradients at Sub-levels of Average Energy Silhouette Images**. *IEEE Transactions on Cognitive and Developmental Systems*, 2016. 15
- [35] SWARUP KUMAR DHAR, MD MAHMUDUL HASAN, AND SHAYHAN AMEEN CHOWDHURY. **Human activity recognition based on Gaussian mixture model and directive local binary pattern**. In *Electrical, Computer & Telecommunication Engineering (ICECTE), International Conference on*, pages 1–4. IEEE, 2016. 15
- [36] SABANADESAN UMAKANTHAN, SIMON DENMAN, CLINTON FOOKES, AND SRIDHA SRIDHARAN. **Multiple instance dictionary learning for activity representation**. In *Pattern Recognition (ICPR), 2014 22nd International Conference on*, pages 1377–1382. IEEE, 2014. 15
- [37] HUIQUAN ZHANG AND OSAMU YOSHIE. **Improving human activity recognition using subspace clustering**. In *Machine Learning and Cybernetics (ICMLC), 2012 International Conference on*, **3**, pages 1058–1063. IEEE, 2012. 16, 88
- [38] NEIL ROBERTSON AND IAN REID. **A general method for human activity recognition in video**. *Computer Vision and Image Understanding*, **104**(2):232–248, 2006. 16
- [39] PAVAN TURAGA, RAMA CHELLAPPA, VENKATRAMANA S SUBRAHMANIAN, AND OCTAVIAN UDREA. **Machine recognition of human activities: A survey**. *IEEE Transactions on Circuits and Systems for Video Technology*, **18**(11):1473–1488, 2008. 16, 57
- [40] TIAN LAN, YANG WANG, WEILONG YANG, AND GREG MORI. **Beyond actions: Discriminative models for contextual group activities**. In *Advances in neural information processing systems*, pages 1216–1224, 2010. 17, 25

-
- [41] MOEZ BACCOUCHE, FRANCK MAMALET, CHRISTIAN WOLF, CHRISTOPHE GARCIA, AND ATILLA BASKURT. **Action classification in soccer videos with long short-term memory recurrent neural networks.** In *International Conference on Artificial Neural Networks*, pages 154–159. Springer, 2010. 17, 27
- [42] MOEZ BACCOUCHE, FRANCK MAMALET, CHRISTIAN WOLF, CHRISTOPHE GARCIA, AND ATILLA BASKURT. **Sequential deep learning for human action recognition.** In *International Workshop on Human Behavior Understanding*, pages 29–39. Springer, 2011. 17, 27
- [43] JOSHUA SUSSKIND, VOLODYMYR MNIH, GEOFFREY HINTON, ET AL. **On deep generative models with applications to recognition.** In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2857–2864. IEEE, 2011. 17, 29
- [44] YICHUAN TANG, RUSLAN SALAKHUTDINOV, AND GEOFFREY HINTON. **Robust boltzmann machines for recognition and denoising.** In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2264–2271. IEEE, 2012. 17, 29
- [45] TIAN LAN, LEONID SIGAL, AND GREG MORI. **Social roles in hierarchical models for human activity recognition.** In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1354–1361. IEEE, 2012. 17, 25
- [46] SHUIWANG JI, WEI XU, MING YANG, AND KAI YU. **3D convolutional neural networks for human action recognition.** *IEEE transactions on pattern analysis and machine intelligence*, **35**(1):221–231, 2013. 17, 27, 30
- [47] ALEX KRIZHEVSKY, ILYA SUTSKEVER, AND GEOFFREY E HINTON. **Imagenet classification with deep convolutional neural networks.** In *Advances in neural information processing systems*, pages 1097–1105, 2012. 17, 30, 33
- [48] KAREN SIMONYAN AND ANDREW ZISSERMAN. **Two-stream convolutional networks for action recognition in videos.** In *Advances in Neural Information Processing Systems*, pages 568–576, 2014. 17, 27

-
- [49] JÜRGEN SCHMIDHUBER. **Deep learning in neural networks: An overview.** *Neural Networks*, **61**:85–117, 2015. 17, 19
- [50] ANDREJ KARPATHY, GEORGE TODERICI, SANKETH SHETTY, THOMAS LEUNG, RAHUL SUKTHANKAR, AND LI FEI-FEI. **Large-scale Video Classification with Convolutional Neural Networks.** 2014. 17, 27
- [51] KAREN SIMONYAN AND ANDREW ZISSERMAN. **Very deep convolutional networks for large-scale image recognition.** *arXiv preprint arXiv:1409.1556*, 2014. 17
- [52] HONGQING FANG AND CHEN HU. **Recognizing human activity in smart home using deep learning algorithm.** In *Control Conference (CCC), 2014 33rd Chinese*, pages 4716–4720. IEEE, 2014. 17, 24
- [53] HAILONG LIU AND TADAHIRO TANIGUCHI. **Feature extraction and pattern recognition for human motion by a deep sparse autoencoder.** In *Computer and Information Technology (CIT), 2014 IEEE International Conference on*, pages 173–181. IEEE, 2014. 17, 24, 28
- [54] SM ALI ESLAMI, NICOLAS HEES, CHRISTOPHER KI WILLIAMS, AND JOHN WINN. **The shape boltzmann machine: a strong model of object shape.** *International Journal of Computer Vision*, **107**(2):155–176, 2014. 17, 29
- [55] ZHIWEI DENG, MENGGAO ZHAI, LEI CHEN, YUHAO LIU, SRIKANTH MURALIDHARAN, MEHRAN JAVAN ROSHTKHARI, AND GREG MORI. **Deep structured models for group activity recognition.** *arXiv preprint arXiv:1506.04191*, 2015. 17, 25
- [56] MOUSTAFA IBRAHIM, SRIKANTH MURALIDHARAN, ZHIWEI DENG, ARASH VAHDAT, AND GREG MORI. **A Hierarchical Deep Temporal Model for Group Activity Recognition.** *arXiv preprint arXiv:1511.06040*, 2015. 17, 25, 27
- [57] CHRISTIAN SZEGEDY, WEI LIU, YANGQING JIA, PIERRE SERMANET, SCOTT REED, DRAGOMIR ANGUELOV, DUMITRU ERHAN, VINCENT VANHOUCHE, AND ANDREW RABINOVICH. **Going deeper with convolutions.** In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 17, 19, 26

-
- [58] JEFFREY DONAHUE, LISA ANNE HENDRICKS, SERGIO GUADARRAMA, MARCUS ROHRBACH, SUBHASHINI VENUGOPALAN, KATE SAENKO, AND TREVOR DARRELL. **Long-term recurrent convolutional networks for visual recognition and description.** In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015. 17, 26, 29
- [59] MAHMUDUL HASAN AND AMIT K ROY-CHOWDHURY. **A continuous learning framework for activity recognition using deep hybrid feature models.** *IEEE Transactions on Multimedia*, **17**(11):1909–1922, 2015. 17, 28
- [60] KATERINA FRAGKIADAKI, SERGEY LEVINE, PANNA FELSEN, AND JITENDRA MALIK. **Recurrent network models for human dynamics.** In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4346–4354, 2015. 17, 29
- [61] SHUAI ZHENG, SADEEP JAYASUMANA, BERNARDINO ROMERA-PAREDES, VIBHAV VINEET, ZHIZHONG SU, DALONG DU, CHANG HUANG, AND PHILIP HS TORR. **Conditional random fields as recurrent neural networks.** In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015. 17, 29
- [62] ASHESH JAIN, AMIR R ZAMIR, SILVIO SAVARESE, AND ASHUTOSH SAXENA. **Structural-RNN: Deep Learning on Spatio-Temporal Graphs.** *arXiv preprint arXiv:1511.05298*, 2015. 17, 29, 88
- [63] ZHIWEI DENG, ARASH VAHDAT, HEXIANG HU, AND GREG MORI. **Structure Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition.** *arXiv preprint arXiv:1511.04196*, 2015. 17, 26
- [64] PICHAO WANG, WANQING LI, ZHIMIN GAO, JING ZHANG, CHANG TANG, AND PHILIP O OGUNBONA. **Action recognition from depth maps using deep convolutional neural networks.** *IEEE Transactions on Human-Machine Systems*, **46**(4):498–509, 2016. 17, 20
- [65] XIAOCHUAN YIN AND QIJUN CHEN. **Deep metric learning autoencoder for nonlinear temporal alignment of human motion.** In *Robotics and*

-
- Automation (ICRA), 2016 IEEE International Conference on*, pages 2160–2166. IEEE, 2016. 17, 21, 24, 28
- [66] TIANLIANG LIU, XINCHENG WANG, XIUBIN DAI, AND JIEBO LUO. **Deep recursive and hierarchical conditional random fields for human action recognition.** In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–9. IEEE, 2016. 17, 30
- [67] XUDONG JIANG, BAPPADITYA MANDAL, AND ALEX KOT. **Eigenfeature regularization and extraction in face recognition.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(3):383–394, 2008. 16, 60, 62
- [68] BABACK MOGHADDAM, TONY JEBARA, AND ALEX PENTLAND. **Bayesian face recognition.** *Pattern Recognition*, **33**(11):1771–1782, 2000. 16
- [69] FESTUS OSAYAMWEN AND JULES R TAPAMO. **Within-class subspace regularization for human activity recognition.** In *2016 IEEE International Conference on Industrial Technology (ICIT)*, pages 804–807. IEEE, 2016. 16, 97
- [70] RONALD POPPE. **A survey on vision-based human action recognition.** *Image and vision computing*, **28**(6):976–990, 2010. 18
- [71] HA JHUANG, HA GARROTE, EA POGGIO, TA SERRE, AND T HMDB. **A large video database for human motion recognition.** In *Proc. of IEEE International Conference on Computer Vision*, 2011. 18
- [72] HUA YU AND JIE YANG. **A direct LDA algorithm for high-dimensional data?with application to face recognition.** *Pattern recognition*, **34**(10):2067–2070, 2001. 18
- [73] LOTHAR REICHEL, FIORELLA SGALLARI, AND QIANG YE. **Tikhonov regularization based on generalized Krylov subspace methods.** *Applied Numerical Mathematics*, **62**(9):1215–1228, 2012. 18
- [74] PIERRE BALDI AND PETER SADOWSKI. **The dropout learning algorithm.** *Artificial intelligence*, **210**:78–122, 2014. 19

-
- [75] CHRISTIAN SZEGEDY, ALEXANDER TOSHEV, AND DUMITRU ERHAN. **Deep neural networks for object detection**. In *Advances in Neural Information Processing Systems*, pages 2553–2561, 2013. 19, 20
- [76] YANN LECUN, YOSHUA BENGIO, AND GEOFFREY HINTON. **Deep learning**. *Nature*, **521**(7553):436–444, 2015. 20, 21
- [77] RUSLAN SALAKHUTDINOV, JOSHUA B TENENBAUM, AND ANTONIO TORRALBA. **Learning with hierarchical-deep models**. *IEEE transactions on pattern analysis and machine intelligence*, **35**(8):1958–1971, 2013. 20
- [78] HANLIN GOH, NICOLAS THOME, MATTHIEU CORD, AND JOO-HWEE LIM. **Learning deep hierarchical visual feature coding**. *IEEE transactions on neural networks and learning systems*, **25**(12):2212–2225, 2014. 20
- [79] BORISLAV ANTIC AND BJÖRN OMMER. **Learning latent constituents for recognition of group activities in video**. In *European Conference on Computer Vision*, pages 33–47. Springer, 2014. 20, 26, 88, 97
- [80] KUMPATI S NARENDRA AND MANDAYAM AL THATHACHAR. **Learning automata-a survey**. *IEEE Transactions on systems, man, and cybernetics*, (4):323–334, 1974. 20
- [81] FRANK ROSENBLATT. **The perceptron: A probabilistic model for information storage and organization in the brain**. *Psychological review*, **65**(6):386, 1958.
- [82] BERNARD WIDROW AND MARCIAN E HOFF. **Associative Storage and Retrieval of Digital Information in Networks of Adaptive ?Neurons?** In *Biological Prototypes and Synthetic Systems*, pages 160–160. Springer, 1962.
- [83] STEPHEN GROSSBERG. **Adaptive pattern classification and universal recoding: II. Feedback, expectation, olfaction, illusions**. *Biological cybernetics*, **23**(4):187–202, 1976. 20
- [84] GOTTFRIED WILHELM LEIBNIZ. *Nova methodus pro maximus et minimus itemque tangentibus: quae nec fractas nec irrationales quantitates moratur, et singulare pro illis calculi genus*. 1884. 20

-
- [85] JACQUES HADAMARD. *Mémoire sur le problème d'analyse relatif à l'équilibre des plaques élastiques encastrées*, **33**. Imprimerie nationale, 1908.
- [86] LEV SEMENOVICH PONTRYAGIN. *Mathematical theory of optimal processes*. CRC Press, 1987. 20
- [87] XAVIER GLOROT AND YOSHUA BENGIO. **Understanding the difficulty of training deep feedforward neural networks**. In *Aistats*, **9**, pages 249–256, 2010. 4, 21, 32
- [88] YOSHUA BENGIO. **Learning deep architectures for AI**. *Foundations and trends® in Machine Learning*, **2(1)**:1–127, 2009. 21, 23
- [89] GEOFFREY E HINTON, SIMON OSINDERO, AND YEE-WHYE TEH. **A fast learning algorithm for deep belief nets**. *Neural computation*, **18(7)**:1527–1554, 2006. 21, 23, 24
- [90] GEOFFREY E HINTON AND RUSLAN R SALAKHUTDINOV. **Reducing the dimensionality of data with neural networks**. *Science*, **313(5786)**:504–507, 2006. 21, 28
- [91] CHRISTINE CHIN AND DAVID E BROWN. **Learning in science: A comparison of deep and surface approaches**. *Journal of research in science teaching*, **37(2)**:109–138, 2000. 21
- [92] HONGLAK LEE, ROGER GROSSE, RAJESH RANGANATH, AND ANDREW Y NG. **Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations**. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616. ACM, 2009.
- [93] DAN CIREGAN, UELI MEIER, AND JÜRGEN SCHMIDHUBER. **Multi-column deep neural networks for image classification**. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012. 21, 90
- [94] JOHAN HASTAD. **Almost optimal lower bounds for small depth circuits**. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 6–20. ACM, 1986. 23

-
- [95] JOHAN HÅSTAD AND MIKAEL GOLDMANN. **On the power of small-depth threshold circuits.** *Computational Complexity*, **1**(2):113–129, 1991.
- [96] YOSHUA BENGIO, OLIVIER DELALLEAU, AND NICOLAS LE ROUX. **The curse of highly variable functions for local kernel machines.** *Advances in neural information processing systems*, **18**:107, 2006.
- [97] YOSHUA BENGIO, YANN LECUN, ET AL. **Scaling learning algorithms towards AI.** *Large-scale kernel machines*, **34**(5):1–41, 2007.
- [98] YOSHUA BENGIO AND MARTIN MONPERRUS. **Non-Local Manifold Tangent Learning.** In *NIPS*, pages 129–136, 2004.
- [99] YOSHUA BENGIO AND OLIVIER DELALLEAU. **On the expressive power of deep architectures.** In *International Conference on Algorithmic Learning Theory*, pages 18–36. Springer, 2011. 1, 23
- [100] YOSHUA BENGIO, PASCAL LAMBLIN, DAN POPOVICI, HUGO LAROCHELLE, ET AL. **Greedy layer-wise training of deep networks.** *Advances in neural information processing systems*, **19**:153, 2007. 23
- [101] CHRISTOPHER POULTNEY MARC?AURELIO RANZATO, SUMIT CHOPRA, AND YANN LECUN. **Efficient learning of sparse representations with an energy-based model.** In *Proceedings of NIPS*, 2007. 23
- [102] HONGLAK LEE, CHAITANYA EKANADHAM, AND ANDREW Y NG. **Sparse deep belief net model for visual area V2.** In *Advances in neural information processing systems*, pages 873–880, 2008. 23
- [103] HUGO LAROCHELLE, YOSHUA BENGIO, JÉRÔME LOURADOUR, AND PASCAL LAMBLIN. **Exploring strategies for training deep neural networks.** *Journal of Machine Learning Research*, **10**(Jan):1–40, 2009. 24
- [104] DUMITRU ERHAN, YOSHUA BENGIO, AARON COURVILLE, PIERRE-ANTOINE MANZAGOL, PASCAL VINCENT, AND SAMY BENGIO. **Why does unsupervised pre-training help deep learning?** *Journal of Machine Learning Research*, **11**(Feb):625–660, 2010. 24, 32

-
- [105] RUSLAN SALAKHUTDINOV AND GEOFFREY E HINTON. **Deep Boltzmann Machines**. In *AISTATS*, **1**, page 3, 2009. 24
- [106] IAN GOODFELLOW, HONGLAK LEE, QUOC V LE, ANDREW SAXE, AND ANDREW Y NG. **Measuring invariances in deep networks**. In *Advances in neural information processing systems*, pages 646–654, 2009. 24
- [107] JULIAN BESAG. **Statistical analysis of non-lattice data**. *The statistician*, pages 179–195, 1975. 24
- [108] JAMES S BERGSTRA, RÉMI BARDENET, YOSHUA BENGIO, AND BALÁZS KÉGL. **Algorithms for hyper-parameter optimization**. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011. 25
- [109] TIAN LAN, YANG WANG, WEILONG YANG, STEPHEN N ROBINOVITCH, AND GREG MORI. **Discriminative latent models for recognizing contextual group activities**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(8):1549–1562, 2012. 26
- [110] MIN LIN, QIANG CHEN, AND SHUICHENG YAN. **Network in network**. *arXiv preprint arXiv:1312.4400*, 2013. 26
- [111] MOHAMED RABIE AMER, PENG LEI, AND SINISA TODOROVIC. **Hirf: Hierarchical random field for collective activity recognition in videos**. In *European Conference on Computer Vision*, pages 572–585. Springer, 2014. 26
- [112] HILDEGARD KUEHNE, HUEIHAN JHUANG, ESTÍBALIZ GARROTE, TOMASO POGGIO, AND THOMAS SERRE. **HMDB: a large video database for human motion recognition**. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. 27
- [113] KHURRAM SOOMRO, AMIR ROSHAN ZAMIR, AND MUBARAK SHAH. **UCF101: A dataset of 101 human actions classes from videos in the wild**. *arXiv preprint arXiv:1212.0402*, 2012. 27

-
- [114] YUUSUKE KATAOKA, TAKASHI MATSUBARA, AND KUNIAKI UEHARA. **Image generation using generative adversarial networks and attention mechanism.** In *Computer and Information Science (ICIS), 2016 IEEE/ACIS 15th International Conference on*, pages 1–6. IEEE, 2016. 28
- [115] QUOC V LE. **Building high-level features using large scale unsupervised learning.** In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8595–8598. IEEE, 2013. 28, 30
- [116] HENG WANG AND CORDELIA SCHMID. **Action recognition with improved trajectories.** In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3551–3558, 2013. 28
- [117] D RANDALL WILSON AND TONY R MARTINEZ. **The general inefficiency of batch training for gradient descent learning.** *Neural Networks*, **16**(10):1429–1451, 2003. 28
- [118] GEOFFREY E HINTON. **A practical guide to training restricted boltzmann machines.** In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012. 28
- [119] TIJMEN TIELEMAN. **Training restricted Boltzmann machines using approximations to the likelihood gradient.** In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM, 2008.
- [120] RUSLAN SALAKHUTDINOV AND HUGO LAROCHELLE. **Efficient Learning of Deep Boltzmann Machines.** In *AISTATs*, **9**, pages 693–700, 2010. 28
- [121] ABDEL-RAHMAN MOHAMED, GEORGE E DAHL, AND GEOFFREY HINTON. **Acoustic modeling using deep belief networks.** *IEEE Transactions on Audio, Speech, and Language Processing*, **20**(1):14–22, 2012. 29
- [122] ZUOGUAN WANG, GERWIN SCHALK, AND QIANG JI. **Anatomically Constrained Decoding of Finger Flexion from Electrocorticographic Signals.** In *NIPS*, pages 2070–2078, 2011. 29

-
- [123] SIQI NIE, ZIHENG WANG, AND QIANG JI. **A generative restricted Boltzmann machine based method for high-dimensional motion data modeling.** *Computer Vision and Image Understanding*, **136**:14–22, 2015. 29
- [124] FERNANDO J PINEDA. **Generalization of back-propagation to recurrent neural networks.** *Physical review letters*, **59**(19):2229, 1987. 29
- [125] TOMAS MIKOLOV, ARMAND JOULIN, SUMIT CHOPRA, MICHAEL MATHIEU, AND MARC'AURELIO RANZATO. **Learning longer memory in recurrent neural networks.** *arXiv preprint arXiv:1412.7753*, 2014. 29
- [126] QINFENG SHI, LI CHENG, LI WANG, AND ALEX SMOLA. **Human action segmentation and recognition using discriminative semi-markov models.** *International journal of computer vision*, **93**(1):22–32, 2011.
- [127] ILYA SUTSKEVER, ORIOL VINYALS, AND QUOC V LE. **Sequence to sequence learning with neural networks.** In *Advances in neural information processing systems*, pages 3104–3112, 2014. 29
- [128] ALEX GRAVES AND NAVDEEP JAITLEY. **Towards End-To-End Speech Recognition with Recurrent Neural Networks.** In *ICML*, **14**, pages 1764–1772, 2014. 29
- [129] BERTRAND DOUILLARD, DIETER FOX, AND FABIO RAMOS. **A spatio-temporal probabilistic model for multi-sensor multi-class object recognition.** In *Robotics Research*, pages 123–134. Springer, 2010. 29
- [130] ASHESH JAIN, HEMA S KOPPULA, BHARAD RAGHAVAN, SHANE SOH, AND ASHUTOSH SAXENA. **Car that knows before you do: Anticipating maneuvers via learning temporal driving models.** In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3182–3190, 2015.
- [131] YUAN LI AND RAM NEVATIA. **Key Object Driven Multi-category Object Recognition, Localization and Tracking Using Spatio-temporal Context.** In *ECCV (4)*, pages 409–422, 2008.

-
- [132] HEMA SWETHA KOPPULA AND ASHUTOSH SAXENA. **Learning Spatio-Temporal Structure from RGB-D Videos for Human Activity Detection and Anticipation.** In *ICML (3)*, pages 792–800, 2013.
- [133] JOSÉ LEZAMA, KARTEEK ALAHARI, JOSEF SIVIC, AND IVAN LAPTEV. **Track to the future: Spatio-temporal video segmentation with long-range motion cues.** In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011.
- [134] XUETAO ZHANG, PEILIN JIANG, AND FEI WANG. **Overtaking vehicle detection using a spatio-temporal CRF.** In *2014 IEEE Intelligent Vehicles Symposium Proceedings*, pages 338–343. IEEE, 2014. 29
- [135] NAM THANH NGUYEN, DINH Q PHUNG, SVETHA VENKATESH, AND HUNG BUI. **Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model.** In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, **2**, pages 955–960. IEEE, 2005. 29
- [136] EUNJU KIM, SUMI HELAL, AND DIANE COOK. **Human activity recognition and pattern discovery.** *IEEE Pervasive Computing*, **9**(1), 2010. 29
- [137] DOUGLAS L VAIL, MANUELA M VELOSO, AND JOHN D LAFFERTY. **Conditional random fields for activity recognition.** In *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, page 235. ACM, 2007. 29
- [138] CRISTIAN SMINCHISESCU, ATUL KANAUIA, AND DIMITRIS METAXAS. **Conditional models for contextual human motion recognition.** *Computer Vision and Image Understanding*, **104**(2):210–220, 2006. 29
- [139] DONG YU, SHIZHEN WANG, AND LI DENG. **Sequential labeling using deep-structured conditional random fields.** *IEEE Journal of Selected Topics in Signal Processing*, **4**(6):965–973, 2010. 30
- [140] GEORG HEIGOLD, HERMANN NEY, PATRICK LEHNEN, TOBIAS GASS, AND RALF SCHLUTER. **Equivalence of generative and log-linear models.** *IEEE*

-
- Transactions on Audio, Speech, and Language Processing*, **19**(5):1138–1148, 2011. 30
- [141] DONG YU, SHIZHEN WANG, ZAHY KARAM, AND LI DENG. **Language recognition using deep-structured conditional random fields**. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5030–5033. IEEE, 2010. 30
- [142] PING WEI, YIBIAO ZHAO, NANNING ZHENG, AND SONG-CHUN ZHU. **Modeling 4d human-object interactions for event and object recognition**. In *2013 IEEE International Conference on Computer Vision*, pages 3272–3279. IEEE, 2013. 30
- [143] BANGPENG YAO AND LI FEI-FEI. **Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(9):1691–1703, 2012. 30
- [144] GEOFFREY E HINTON, ALEX KRIZHEVSKY, AND SIDA D WANG. **Transforming auto-encoders**. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer, 2011. 30, 90
- [145] YANN LECUN, LÉON BOTTOU, YOSHUA BENGIO, AND PATRICK HAFFNER. **Gradient-based learning applied to document recognition**. *Proceedings of the IEEE*, **86**(11):2278–2324, 1998. 30, 31, 106
- [146] DAN CIRESAN, ALESSANDRO GIUSTI, LUCA M GAMBARDELLA, AND JÜRGEN SCHMIDHUBER. **Deep neural networks segment neuronal membranes in electron microscopy images**. In *Advances in neural information processing systems*, pages 2843–2851, 2012. 30, 97
- [147] LORENZO PORZI, SAMUEL ROTA BULO, ADRIAN PENATE-SANCHEZ, ELISA RICCI, AND FRANCESC MORENO-NOGUER. **Learning Depth-aware Deep Representations for Robotic Perception**. *IEEE Robotics and Automation Letters*, 2016. 31

-
- [148] RAFFAY HAMID, SIDDHARTHA MADDI, AMOS JOHNSON, AARON BOBICK, IRFAN ESSA, AND CHARLES ISBELL. **A novel sequence representation for unsupervised analysis of human activities.** *Artificial Intelligence*, **173**(14):1221–1244, 2009. 31
- [149] BRIAN P BLOOMFIELD AND THEO VURDUBAKIS. **Visions of organization and organizations of vision: the representational practices of information systems development.** *Accounting, Organizations and Society*, **22**(7):639–668, 1997. 31
- [150] YOSHUA BENGIO, JÉRÔME LOURADOUR, RONAN COLLOBERT, AND JASON WESTON. **Curriculum learning.** In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009. 32
- [151] JASON WESTON, FRÉDÉRIC RATLE, HOSSEIN MOBAHI, AND RONAN COLLOBERT. **Deep learning via semi-supervised embedding.** In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012. 32
- [152] TAKAYOSHI YAMASHITA, MASAYUKI TANAKA, YUJI YAMAUCHI, AND HIRONOBU FUJIYOSHI. **SWAP-NODE: A regularization approach for deep convolutional neural networks.** In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2475–2479. IEEE, 2015. 32
- [153] MICHIEL HERMANS AND BENJAMIN SCHRAUWEN. **Training and analysing deep recurrent neural networks.** In *Advances in neural information processing systems*, pages 190–198, 2013. 32
- [154] JAN KOUTNIK, KLAUS GREFF, FAUSTINO GOMEZ, AND JUERGEN SCHMIDHUBER. **A clockwork rnn.** *arXiv preprint arXiv:1402.3511*, 2014. 33
- [155] HERBERT JAEGER AND HARALD HAAS. **Harnessing nonlinearity: prediction of chaotic time series with neural networks.** *Science*, **304**(5667):78–80, 2004.
- [156] RAZVAN PASCANU, TOMAS MIKOLOV, AND YOSHUA BENGIO. **On the difficulty of training recurrent neural networks.** *ICML (3)*, **28**:1310–1318, 2013. 33

-
- [157] JÜRGEN SCHMIDHUBER, DAAN WIERSTRA, MATTEO GAGLIOLO, AND FAUSTINO GOMEZ. **Training recurrent networks by evolino.** *Neural computation*, **19**(3):757–779, 2007. 33
- [158] ILYA SUTSKEVER, GEOFFREY E HINTON, AND GRAHAM W TAYLOR. **The recurrent temporal restricted boltzmann machine.** In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2009. 33
- [159] SHUN-ICHI AMARI. **Natural gradient works efficiently in learning.** *Neural computation*, **10**(2):251–276, 1998. 33
- [160] RAZVAN PASCANU AND YOSHUA BENGIO. **Natural gradient revisited.** Technical report, Technical report, 2013. 33
- [161] NICOLAS L ROUX, PIERRE-ANTOINE MANZAGOL, AND YOSHUA BENGIO. **Topmoutoute online natural gradient algorithm.** In *Advances in neural information processing systems*, pages 849–856, 2008. 33
- [162] KYUNGHYUN CHO, TAPANI RAIKO, AND ALEXANDER T IHLER. **Enhanced gradient and adaptive learning rate for training restricted Boltzmann machines.** In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 105–112, 2011. 33
- [163] XAVIER GLOROT, ANTOINE BORDES, AND YOSHUA BENGIO. **Deep Sparse Rectifier Neural Networks.** In *Aistats*, **15**, page 275, 2011. 33, 83, 90
- [164] VINOD NAIR AND GEOFFREY E HINTON. **Rectified linear units improve restricted boltzmann machines.** In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010. 33, 83
- [165] KEVIN JARRETT, KORAY KAVUKCUOGLU, YANN LECUN, ET AL. **What is the best multi-stage architecture for object recognition?** In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2146–2153. IEEE, 2009. 33
- [166] TAPANI RAIKO, HARRI VALPOLA, AND YANN LECUN. **Deep Learning Made Easier by Linear Transformations in Perceptrons.** In *AISTATS*, **22**, pages 924–932, 2012. 33

-
- [167] DAN CLAUDIU CIREŞAN, UELI MEIER, LUCA MARIA GAMBARDILLA, AND JÜRGEN SCHMIDHUBER. **Deep, big, simple neural nets for handwritten digit recognition.** *Neural computation*, **22**(12):3207–3220, 2010. 33, 90
- [168] FRANK SEIDE, GANG LI, AND DONG YU. **Conversational Speech Transcription Using Context-Dependent Deep Neural Networks.** In *Interspeech*, pages 437–440, 2011. 33
- [169] THIERRY BOUWMANS, FIDA EL BAF, AND BERTRAND VACHON. **Background modeling using mixture of gaussians for foreground detection-a survey.** *Recent Patents on Computer Science*, **1**(3):219–237, 2008. 36
- [170] CHRIS STAUFFER AND W ERIC L GRIMSON. **Adaptive background mixture models for real-time tracking.** In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, **2**, pages 246–252. IEEE, 1999. 36, 37, 38
- [171] NIR FRIEDMAN AND STUART RUSSELL. **Image segmentation in video sequences: A probabilistic approach.** In *Proceedings of the Thirteenth conference on Uncertainty in artificial intelligence*, pages 175–181. Morgan Kaufmann Publishers Inc., 1997. 37
- [172] STEFAN ATEV, OSAMA MASOUD, AND NIKOS PAPANIKOLOPOULOS. **Practical mixtures of Gaussians with brightness monitoring.** In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 423–428. IEEE, 2004. 39
- [173] QI ZANG AND REINHARD KLETTE. **Parameter analysis for mixture of gaussians model.** Technical report, CITR, The University of Auckland, New Zealand, 2006. 39
- [174] CHUAN SUN, IMRAN JUNEJO, AND HASSAN FOROOSH. **Action recognition using rank-1 approximation of joint self-similarity volume.** In *2011 International Conference on Computer Vision*, pages 1007–1012. IEEE, 2011. 40, 47

-
- [175] SALVATORE TABBONE, LAURENT WENDLING, AND J-P SALMON. **A new shape descriptor defined on the Radon transform.** *Computer Vision and Image Understanding*, **102**(1):42–51, 2006.
- [176] FENG NIU AND MOHAMED ABDEL-MOTTALEB. **View-invariant human activity recognition based on shape and motion features.** In *Multimedia Software Engineering, 2004. Proceedings. IEEE Sixth International Symposium on*, pages 546–556. IEEE, 2004. 47, 48
- [177] FENG NIU AND MOHAMED ABDEL-MOTTALEB. **HMM-based segmentation and recognition of human activities from video sequences.** In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 804–807. IEEE, 2005. 48
- [178] JINFENG BAI, ZHINENG CHEN, BAILAN FENG, AND BO XU. **Chinese image text recognition on grayscale pixels.** In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 1380–1384. IEEE, 2014. 48
- [179] BINGBING NI, YONG PEI, PIERRE MOULIN, AND SHUICHENG YAN. **Multilevel depth and image fusion for human activity detection.** *IEEE transactions on cybernetics*, **43**(5):1383–1394, 2013. 49
- [180] SURYANI ILIAS, NOORITAWATI MD TAHIR, AND ROZITA JAILANI. **Feature extraction of autism gait data using principal component analysis and linear discriminant analysis.** In *Industrial Electronics and Applications Conference (IEACon), 2016 IEEE*, pages 275–279. IEEE, 2016. 58
- [181] YAN-MING ZHANG, XINWEN HOU, SHIMING XIANG, AND CHENG-LIN LIU. **Subspace regularization: A new semi-supervised learning method.** In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 586–601. Springer, 2009. 59
- [182] YINGYING ZHU, NANDITA M NAYAK, AND AMIT K ROY-CHOWDHURY. **Context-aware modeling and recognition of activities in video.** In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2491–2498, 2013. 62

-
- [183] YIMENG ZHANG, XIAOMING LIU, MING-CHING CHANG, WEINA GE, AND TSUHAN CHEN. **Spatio-temporal phrases for activity recognition**. In *European Conference on Computer Vision*, pages 707–721. Springer, 2012.
- [184] MM SARDSEHMUKH, MT KOLTE, PN CHATUR, AND DS CHAUDHARI. **3-D dataset for Human Activity Recognition in video surveillance**. In *Wireless Computing and Networking (GCWCN), 2014 IEEE Global Conference on*, pages 75–78. IEEE, 2014. 62
- [185] HUSSEIN MAZAAR, EID EMARY, AND HODA ONSI. **Evaluation of feature selection on human activity recognition**. In *2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS)*, pages 591–599. IEEE, 2015. 64, 125
- [186] WEIYAO LIN, YUANZHE CHEN, JIANXIN WU, HANLI WANG, BIN SHENG, AND HONGXIANG LI. **A new network-based algorithm for human activity recognition in videos**. *IEEE Transactions on Circuits and Systems for Video Technology*, **24**(5):826–841, 2014. 64
- [187] YANGDA ZHU, CHANGHAI WANG, JIANZHONG ZHANG, AND JINGDONG XU. **Human activity recognition based on similarity**. In *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*, pages 1382–1387. IEEE, 2014. 66
- [188] YI SUN, XIAOGANG WANG, AND XIAOOU TANG. **Deeply learned face representations are sparse, selective, and robust**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2892–2900, 2015. 83
- [189] YANDONG WEN, KAIPENG ZHANG, ZHIFENG LI, AND YU QIAO. **A discriminative feature learning approach for deep face recognition**. In *European Conference on Computer Vision*, pages 499–515. Springer, 2016. 88, 89, 94, 95, 98, 103
- [190] RAIJA HADSELL, SUMIT CHOPRA, AND YANN LECUN. **Dimensionality reduction by learning an invariant mapping**. In *null*, pages 1735–1742. IEEE, 2006. 88

-
- [191] FLORIAN SCHROFF, DMITRY KALENICHENKO, AND JAMES PHILBIN. **Facenet: A unified embedding for face recognition and clustering**. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 88
- [192] FESTUS OSAYAMWEN AND JULES-RAYMOND TAPAMO. **Improved eigenspectrum regularisation for human activity recognition**. *International Journal of Computational Vision and Robotics*, **8**(4):435–454, 2018. 97
- [193] WEITAO WAN, YUANYI ZHONG, TIANPENG LI, AND JIANGSHENG CHEN. **Rethinking feature distribution for loss functions in image classification**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9117–9126, 2018. 98, 107
- [194] TAMIR HAZAN, SUBHRANSU MAJI, JOSEPH KESHET, AND TOMMI JAAKKOLA. **Learning efficient random maximum a-posteriori predictors with non-decomposable loss functions**. In *Advances in Neural Information Processing Systems*, pages 1887–1895, 2013. 99
- [195] STEVEN J NOWLAN AND GEOFFREY E HINTON. **Simplifying neural networks by soft weight-sharing**. *Neural computation*, **4**(4):473–493, 1992. 106
- [196] GEOFFREY E HINTON, NITISH SRIVASTAVA, ALEX KRIZHEVSKY, ILYA SUTSKEVER, AND RUSLAN R SALAKHUTDINOV. **Improving neural networks by preventing co-adaptation of feature detectors**. *arXiv preprint arXiv:1207.0580*, 2012. 106, 115
- [197] ALEX KRIZHEVSKY AND GEOFFREY HINTON. **Learning multiple layers of features from tiny images**. 2009. 106
- [198] MATTHEW D ZEILER AND ROB FERGUS. **Stochastic pooling for regularization of deep convolutional neural networks**. *arXiv preprint arXiv:1301.3557*, 2013. 106
- [199] LI WAN, MATTHEW ZEILER, SIXIN ZHANG, YANN L CUN, AND ROB FERGUS. **Regularization of neural networks using dropconnect**. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1058–1066, 2013. 106

REFERENCES

- [200] NITISH SRIVASTAVA, GEOFFREY HINTON, ALEX KRIZHEVSKY, ILYA SUTSKEVER, AND RUSLAN SALAKHUTDINOV. **Dropout: A simple way to prevent neural networks from overfitting.** *The Journal of Machine Learning Research*, **15**(1):1929–1958, 2014. 115
- [201] HAIBING WU AND XIAODONG GU. **Towards dropout training for convolutional neural networks.** *Neural Networks*, **71**:1–10, 2015. 115
- [202] GUO-CHU CHEN AND JIN-SHOU YU. **Particle swarm optimization algorithm.** *INFORMATION AND CONTROL-SHENYANG-*, **34**(3):318, 2005. 126