

https://dspace.org/

Face detection and recognition in low illumination environment using convolutional neural networks

Bashir, Sirosh

2022
https://hdl.handle.net/10539/34954

Downloaded from a DSpace repository. https://dspace.org/

Face Detection and Recognition in Low Illumination Environments using Convolutional Neural Networks

Author: Sirosh Bashir

A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in fulfillment of the requirements for the degree of Master of Science in Engineering.

Johannesburg,2022

Declaration of Authorship

I declare that this dissertation is my own unaided work except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted for any degree or examination to any other university.



Signed

August/05/2022

Date

Summary

Face detection and recognition in real-time video with low illumination have become important for various applications such as smart cities, airports, underground mines (access control, rescue operations, etc.). However, most of the legacy computer vision techniques are not been able to detect and recognize faces from such images with acceptable accuracy. There is, therefore, a need to explore machine learning techniques to achieve face detection and recognition in low illumination images with acceptable accuracy.

Face detection, registration, and recognition in unconstrained environments have been popular research fields and this has resulted in numerous new methodologies for face detection and recognition in the recent past. One of the most popular techniques in this regard which performs well with good illumination images are the method based on Convolutional Neural Network with standard configurations. Although ample amount of work has been done but low illumination research was not well addressed in the literature review[1][2]. Also, most of the CNN based techniques provide promising results for still images[3]. Thus, there is a need to understand the CNN configuration for low illumination videos.

The research aims to explore the potential of different machine learning approaches for face detection and recognition in a low illumination environment. Different machine learning approaches such as YOLO and LBPH with KNN were used to analyze the accuracy of detection and recognition in a low illumination environment. A number of techniques were applied to improve the detection and recognition accuracy in low illumination. For example, data augmentation was implemented to increase the diversity of the dataset which was later on used for

training purposes. Similarly, preprocessing techniques were applied to a standard dataset to improve the accuracy of the system. The exploratory data analysis technique runs on a preprocessed annotated images dataset. Exploratory data analysis (EDA) gives insight from the dataset, which assists in hyperparameter tuning for training and testing purpose[4]. The simulation results shows that YOLO outperforms with LBPH with KNN for face detection and recognition in a low illumination environment.

Dedication

I dedicate this dissertation to my family and friends. A special feeling of gratitude to my loving parents, whose words of encouragement and push for tenacity ring in my ears. To my brothers and sister, who have never left my side and are very special. Lastly, I dedicate this dissertation to my many friends who supported me throughout the process, especially Dr. Feroze for helping me develop my technical and writing skills.

Acknowledgements

Profound gratitude and appreciation to the following individuals for their guidance, support and encouragement in making this study possible.

- My supervisors, Ms Ellen De Mello Koch, Dr. Asad Mahmood and Prof Ling Cheng for their valuable academic support, guidance and patience throughout this study.
- Dr. Feroze for his support on every step of research.
- Wits Mining Institute, Digital Mining Laboratory and Sibanye-Stillwater for their financial support.
- Dr. Ali and Dr. Hamid Ashraf for their motivation, support and guidance throughout this research.
- All staff members of School of Electrical Engineering and School of Mining Engineering, University of Witwatersrand, for their moral support throughout the stay.

Contents

Abstract **ii**

Dedication **iv**

Acknowledgements **v**

1 Introduction **1**

1.1 Introduction 1

1.2 Motivation 2

1.3 Problem Statement 3

1.4 Research Objectives 3

1.5 Research Questions 4

1.6 Structure of the Dissertation 5

1.6.1 Chapter 1 - Introduction 6

1.6.2 Chapter 2 - Literature Review 6

1.6.3 Chapter 3 - Machine learning Techniques for Face Detection and Recognition 6

1.6.4 Chapter 4 - Low illumination - Problem in face detection and recognition . . . 6

1.6.5 Chapter 5 - Research Methodology . . . 7

1.6.6 Chapter 6 - Implementation of the Algorithm for Face Detection and Recognition . . . 7

1.6.7 Chapter 7 - Results and Discussion . . . 7

1.6.8 Chapter 8 - Conclusions, recommendation and future work . 7

2 Literature Review 8

2.1 Introduction . . . 8

2.2 Methods of Face detection and Recognition . . . 8

2.2.1 Local Approaches . . . 9

(a) Local Appearance-Based Techniques . . . 10

(b) Key-Point Based Techniques . . . 11

2.2.2 Holistic Approaches . . . 11

a) Statistical based approach . . . 12

b) Artificial Intelligence based approach . . . 12

2.2.3 Hybrid Approaches . . . 17

2.2.4 Mostly used Face detection and Recognition Techniques . . . 17

2.2.5 Face Detection and Recognition in Low Illumination . . . 20

2.3 Conclusion . . . 20

3 Machine learning Techniques for Face Detection and Recognition 22

3.1 Introduction 22

3.2 KNN with LBPH for Face Detection and Recognition 23

a) Local Binary Pattern 23

3.3 Convolutional Neural Networks for Face Detection and Recognition . 27

3.4 You Only Look Once(YOLO) 31

3.4.1 Network Design of YOLO 33

(a) Loss Function 33

3.5 Conclusion 36

4 Low illumination Dataset Problem in Face detection and Recognition 37

4.1 Introduction 37

4.2 What is Illumination? 37

4.3 Datasets for Face Detection and Recognition 40

4.3.1 FERET Dataset 40

4.3.2 ORL Database of Faces 40

4.3.3 Extended Yale Face Database B 41

4.4 Conclusion 42

5 Research Methodology 44

5.1 Introduction 44

5.2 Methodology for Proposed approaches 44

5.2.1 Significance of Data for Machine Learning 50

5.2.2 Implementation of models 51

a) Deep Neural Network 52

b) Modification in CNN 53

5.2.3 Databases 56

5.2.4 Accuracy Metrics in Machine Learning 56

5.3 Conclusion 59

6 Implementation of Machine Learning Algorithms for Face Detection and Recognition 60

6.1 Introduction 60

6.2 Extended Yale Dataset B 61

6.2.1 Classification of Extended YaleB Dataset 63

6.2.2 Data Augmentation 66

6.3 Implementation of LBPH with KNN 67

6.4 Implementation of YOLO 69

6.4.1 Data Augmentation for YOLO Implementation 72

6.4.2 Hyperparameters for YOLO Implementation 73

6.4.3 Exploratory Data Analysis 74

6.4.4 Selection of Anchor Boxes 76

6.5 Conclusion 77

7 Results and Discussion **79**

7.1 Introduction 79

7.2 LBPH with KNN using ORL Dataset 79

7.3 LBPH with KNN using the Extended YaleB Dataset 81

7.3.1 Results for Subset 1 81

7.3.2 Results for Subset 2 82

7.3.3 Results for Subset 3 83

7.3.4 Results for Subset 4 84

7.3.5 Results for Subset 5 85

7.4 YOLO Face Detection and Recognition Results 86

7.5 Results for Preprocessed images using CLAHE 90

7.6 Critical Analysis 90

7.7 Conclusion 95

8 Conclusion recommendation and future work **96**

8.1 Conclusion 96

8.2 Recommendations 97

8.3 Future work 97

List of Figures

1.1 Research framework 5

2.1 Classification of methods and techniques for face detection and recognition 9

2.2 Holistic face recognition methods 12

2.3 classification of Neural Networks based Object Detection techniques 13

2.4 Speed (ms) versus accuracy (mAP) [5] 14

2.5 Detection performance under different IoU threshold for each detector when predicting. (a) Faster RCNN(VGG-16); (b) Faster RCNN(ResNet-101); (c) R-FCN(ResNet-101); (d) SSD(InceptionV2); (e) YOLOv3 (Darknet53) [5]. 15

3.1 Basic LBP operator 24

3.2 The circular (8,1), (16,2), and (8,2) neighborhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel[6] 24

3.3 Workflow of Input layer in Convolutional Neural Network 28

3.4 Structure of Convolutional Layer 29

3.5 Schematic representation of max and average Pooling with filter 2×2 with stride 2 30

3.6 Convolutional neural network structure 31

4.1 Measurement of light 38

4.2 Part of the face images from the Feret database 41

4.3 The ORL database for training and testing 42

4.4 Five subset of Extended YaleB dataset 43

5.1 Research methodology 45

5.2 Local binary Pattern Histogram (LBPH) calculation for feature extraction 46

5.3 Ratio of samples and recognition[7][8][9] 47

5.4 Region based Neural network Illustration 48

5.5 Convolutional neural network based approach 51

5.6 YOLO Architecture [7] 53

5.7 Face images from the Extended Yale Face Database B database under different illuminations. 57

5.8 Confusion matrix 58

6.1 Configuration of nine light source directions with all 200 sample in Extended Yale dataset B [10] 62

6.2 Different angle of lighting on subjects of YaleB dataset 62

6.3 Histogram and CDF for (a)Subset 1, (b)Subset 2, (c)Subset 3, (d)Subset 4, (e)Subset 5 64

6.4 Different illumination subsets in the Extended YaleB face database. 65

6.5 Data Augmentation for subject 1 67

6.6 LBPH and KNN implementation 68

6.7 Pre-processing of image using Haar-Cascade Classifier 69

6.8 Framework of Yolo implementation 71

6.9 Bounding boxes on subject images 72

6.10 Plotting data points of training dataset 74

6.11 Kernel density estimation of data points 75

6.12 K mean clustering using k=5 76

7.1 Accuracy with different distance method in KNN 80

7.2 Recognition accuracy of LBPH with KNN with different K values . 82

7.3 Recognition accuracy of LBPH with KNN where angle of light is 0 to 12 degrees(which corresponds to a well illuminated image.) 83

7.4 Recognition accuracy result of subset 2 using LBPH and KNN . . . 84

7.5 Subset 3 Recognition accuracy result for LBPH with KNN 85

7.6 Accuracy results for subset 4 using LBPH with KNN 86

7.7 Low illumination images with 71° to 130°, Subset 5 result using Classical method(LBPH with KNN) 87

7.8 Results of LBPH with KNN considering all subsets 88

7.9 Results of YOLO on different subsets 89

7.10 Histogram of subset 5, subject 1 91

7.11 Histogram of subset 5, After applying CLAHE on subject 1 92

7.12 Recognition accuracy for subset 5 images, with and without CLAHE. 93

7.13 Accuracy on comparison of CLAHE and normal subset of low illumination 94

7.14 Combined results for a)LBPH with KNN 2)YOLO 3) YOLO using Clahe for preprocessing 95

List of Tables

2.1 Confusion matrix 19

3.1 LBP+KNN Recognition accuracy with different distance method . . 26

3.2 CPU and GPU time used by four CNN approaches 35

4.1 Country Based Standards for Illumination in workplace 39

5.1 Performance comparison of YOLO and R-CNN models 46

6.1 Hyperparameter for YOLO 77

7.1 The recognition rate using LBPH and KNN on ORL dataset 80

7.2 YOLO result on Extended YaleB datset 88

7.3 Accuracy of LBPH with KNN and proposed method 92

Acronyms

CLAHE	Contrast-Limited Adaptive Histogram equalization.
CNN	Convolutional Neural Network.
GDP	Gross Domestic Product.
GPU	Graphical Processing Unit.
KNN	K-nearest Neighbors.
LBPH	Local Binary Patter Histogram.
LDA	Linear Discriminant Analysis.
NN	Neural Network.
PCA	Principal Component Analysis.
Relu	Rectified Linear Unit.
YOLO	You Only Look Once.

Chapter 1

Introduction

1.1 Introduction

The mining industry is one of the most significant contributors to the GDP (Gross Domestic Product) of South Africa with more than a million people employed within the industry. There are more than 50 major mines in South Africa, mostly underground, containing minerals such as coal, gold, platinum, iron, diamond, and manganese. The mines, though have become safer with the introduction of well-developed mining practices, still have to combat emergency situations every miner is required to be equipped with the specified protective and emergency equipment. Furthermore, every mine uses access control systems that include both the physical security check, and barcoded cards to prevent unauthorized entries in a mine. However, physical security check points can be bypassed and barcoded cards are duplicable and exchangeable which allows entry of illegal miners causing financial loses to the mine owners and possible risk to the life of the unauthorized miner [11][12]. The security check is not only important for the prevention of unauthorized entry but it is also important to know who is present in a mine in the case of a rescue operation followed after a disaster/accident.

A common method being used is face detection, and recognition applications (computer vision techniques) such as surveillance, and face tagging used by many organizations and security agencies for access control and identification

without human interference [13]. However, the algorithms used for these applications, such as K-nearest neighbor, decision trees, and Viola-Jones have either less accuracy or long processing time which makes these algorithms unsuitable for real-time usage. Furthermore, a typical mining environment may have low illumination of 20 lux which is lesser than the normal lighting conditions typically found in offices which ranges from 200-520 lux [14], resulting in low contrast images for which these algorithms are not suitable. This study would compare the performance of state-of-the-art algorithms in a low-light environment while taking into account the speed at which the algorithm operates.

Real time (in a video sequence) face detection and recognition of low contrast images taken in a low illumination environment is required for various smart city and underground mining (access control, rescue operations etc.) applications. However, it is a real challenge to achieve acceptable accuracy of face detection and recognition in a low illumination environment. The study therefore, aims to provide a step forward for face recognition and detection in low illuminated environments and see how the selection of hyperparameters in convolutional neural network (CNN) can contribute to the improvement of the system's accuracy in low illumination environment.

1.2 Motivation

South Africa is a mineral-rich country and the economy of the country and livelihood of thousands of families is associated with the mining industry. Risk in the mining industry is very high due to the underground harsh environment, and low light increases the risk of an accident. The underground mining environment has low light because of the dust, and poor reflective surfaces. This low lighting cause problems in normal working as well as also enhances the problems while any rescue mission or disaster. Real time and reliable face detection or recognition in a complex low illumination and accident-prone environment may prevent illegal mining and contribute to successful rescue operations. This will contribute to the detection of humans in low illumination with acceptable accuracy, which is useful

in different search and rescue operations in the case of disaster.

Several industries are using drones for safety and security operations, detection of the face in low illumination help drone technology in finding a person in an underground environment. The outcome of this research will also be useful for designing various operations for smart cities and farming industries. In farming, detection of a human at night will reduce the poaching of cows. Smart cities application for security will be benefited from the outcome of this research.

1.3 Problem Statement

A number of computer vision techniques such as K-nearest neighbors (KNN), principal componenet analysis(PCA) and Viola Jones are currently used to perform facial recognition. However, the accuracy of these algorithms for facial detection and recognition in real time for low illuminous environment needs to be tested and improved on low lighting images[15]. Furthermore, these techniques have mostly been implemented, compared, and evaluated for available dataset using images and not on videos of low light[1] environment. Therefore, there is a need to conduct a comparative study of different existing techniques that have been implemented in order to solve face detection and recognition problems in low illumination. Neural Network (NN) is providing promising results in face detection and recognition, has not yet been applied to low illumination environment, and needs to be evaluated for face detection and recognition in low lighting conditions especially in mining industry.

1.4 Research Objectives

The point of the study is to apply NN for face detection and recognition on a low-light video stream to achieve the following goals:-

[1]minimum light in work area according to South African standard in 20-40 lux.

1. Empirically compare the results offered by the YOLO to the currently face recognition system from the literature (K-nearest Neighbors with LBPH) and conclude which algorithm performs best in the context of face recognition from a video sequence in low illumination environment.

2. Evaluate the results in terms of accuracy and performance (time complexity) after using image enhancement technique.

3. To find suitable hyper-parameters for face detection and recognition in low illumination environment.

4. To design a CNN based system for face detection and recognition for low illumination environment.

1.5 Research Questions

Accurate face identification and recognition with lower computational complexity from a video sequence made in a low illuminated setting is a challenge. The research aims at finding answers to the following questions:-

1. What is the accuracy of face detection and recognition, from a video sequence, of KNN with LBPH, and CNN (YOLO) based algorithms with regard to change in luminosity (illumination[2]) ?

2. How can hyperparameters be selected for a YOLO based approach to achieve higher accuracy in face detection and recognition for low illuimination (20-40lx) images?

Accuracy is the degree of conformance between a measurement of an observable quantity and a recognized standard or specification that indicates the true value of the quantity [16]. The accuracy of the image will be calculated using a confusion matrix. The confusion matrix is generally used as a quantitative means

[2]Luminance is a measure of the amount of light falling on a surface.

to define the precision of the image classification. The strength of the uncertainty matrix is that it defines the degree and severity of error classification.

1.6 Structure of the Dissertation

The conceptual framework of the study has two components. Investigation and implementation of classical and the proposed modification for face detection and recognition. The first method is LBPH with KNN and the second one is proposed CNN based YOLO approach which is shown in 1.1. The adopted framework for methodology is given below.

- Detailed literature analysis on the nature and aims of this study.
- Selection of dataset and divide this on the basis of image illumination.
- Implementation of classicals method for face detection and recognition.
- Implementation of YOLO for face detection and recognition with different hyperparameters.

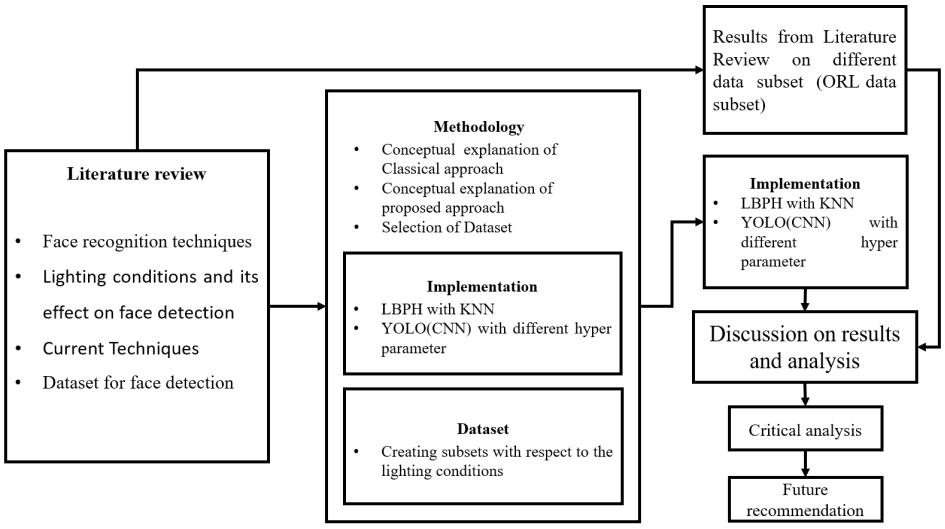


Figure 1.1: Research framework

1.6.1 Chapter 1 - Introduction

The introduction of the problem, the problem statement itself, the nature and objectives of the research, the research questions, the conceptual structure of the research and the organization of the thesis proceeded by the importance of the study are briefly addressed in chapter one.

1.6.2 Chapter 2 - Literature Review

Chapter two presents a review of the various face detection and recognition techniques, information on public datasets and the accuracy of algorithms. This chapter also includes the literature accuracy and contribution of researchers in face detection and recognition.

1.6.3 Chapter 3 - Machine learning Techniques for Face Detection and Recognition

A review of the selected face detection and recognition techniques; the complete structure and working of LBPH with KNN and YOLO is discussed in this chapter. This also includes the information of different layers in Neural networks.

1.6.4 Chapter 4 - Low illumination - Problem in face detection and recognition

This chapter includes the discussion on illumination, the effect of low illumination on detection, and the available datasets for face detection and recognition. Furthermore, the lighting standards and the need/importance of research on face detection and recognition in low lighting conditions has been discussed.

1.6.5 Chapter 5 - Research Methodology

The conceptual framework and methodology implemented for this research are presented in this chapter. The components of the research associated with evaluation of the algorithms are discussed in detail.

1.6.6 Chapter 6 - Implementation of the Algorithm for Face Detection and Recognition

Chapter 5 presents an account of the studies carried out to validate the LBPH with KNN, and CNN based face detection and recognition followed by an explanation of the CLAHE preprocessing method to increase the accuracy of CNN based algorithm.

1.6.7 Chapter 7 - Results and Discussion

The results obtained from the implementation of the algorithms (as covered in chapter (5) are discussed in this chapter.

1.6.8 Chapter 8 - Conclusions, recommendation and future work

This chapter outlines the results of the study, the conclusions taken, the shortcomings of the study and the proposals for future work.

Chapter 2

Literature Review

2.1 Introduction

The chapter would briefly discuss the techniques used to identify and recognize facial images and video sequences, as well as how the identification and recognition mechanism operates. The methods include local approaches, Holistic approaches and Hybrid approach. These methods are further divided into techniques. Lastly, the Deep Neural Network is discussed in the context of object recognition from video and human face recognition from images.

2.2 Methods of Face detection and Recognition

Over the decades, a variety of algorithms and methods have been developed for face detection and recognition. However, the generally used approaches are local approach, Holistic approach, and hybrid approaches, which are further divided in to key point based technique, local appearance based technique, Non linear techniques, and linear techniques. Hybrid approaches are basically the combination of both approaches. A more advanced approach is Convolution Neural Network (CNN) approach. CNN method is one of the main methods adapted in the field of face recognition [17]. Figure 2.1 will describe the classification of face detection and recognition methods and techniques.

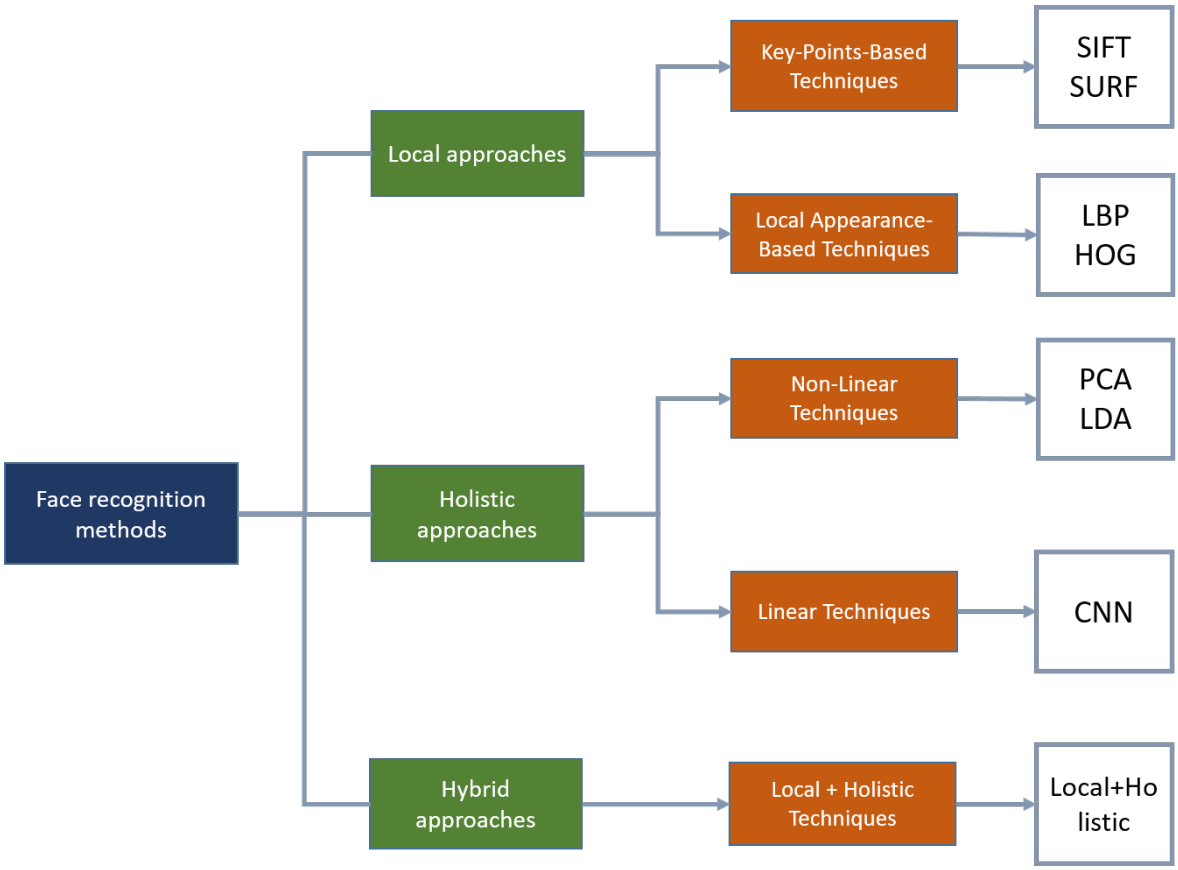


Figure 2.1: Classification of methods and techniques for face detection and recognition

2.2.1 Local Approaches

The feature based or geometrical approaches extract common features of the face from an image, such as the position of the eyes, nose, and mouth and uses the facial coordinates to determine whether two faces are identical or not. A number of techniques can be used for this purpose and one of the simple technique is edge and signatures detection [18]. In the edge detection process, edge maps are extracted and matched with possible variations in size and position. Faces are then matched by searching the features such as eyes and mouth. Generally, these approaches can be divided into two categories: (1) local appearance based techniques are used to extract local features, while the face image is divided into small regions. (2) Key-points-based techniques are used to detect the points of

interest in the face image, after which the features localized to these points are extracted.

(a) Local Appearance-Based Techniques

It is a geometrical technique, also called feature or analytic technique. In this case, the face image is represented by a set of distinctive vectors with low dimensions or small regions (patches). Local appearance-based techniques focus on critical points of the face such as the nose, mouth, and eyes to generate more details. LBP is a great technique to extract features from the object. It has widely performed in many applications which includes face recognition, facial expression recognition, texture segmentation, and texture classification.

Beli et al., Idelette Guo, Chunsheng also used LBPH and KNN [19] but used their own dataset that had 2290 video sequences, which were divided into three sets; training, parametric estimation, and testing. To avoid different face databases, they used a technique in which faces were automatically detected and extracted from the video sequences. The 10 nearest neighbors were chosen for the KNN method and database to achieve a fair trade-off between the identification effects and the processing time.

Kulkarni introduced real-time face detection and recognition using LBPH and KNN [20]. The training dataset consists of 100 training images and one testing image for this study. All images were acquired using the webcam mounted on a ruspburry Pi module. To get the results from this algorithm a value of K=10 was used.

Salim presented a face recognition system for attendance which was based on LBPH with KNN [21]. This system was built on a camera mounted Raspberry Pi. The pictures though were taken and passed to the Raspberry Pi for face recognition, but the feed from the camera to the Raspberry was not live. Low illumination and pose variation decreased the accuracy for this system. This system showed good results when there was only a frontal face picture and light conditions were good. Overall accuracy for 11 subject test database was 95%.

Ahonen proposed an efficient technique for facial image representation based on the local binary pattern texture function. This methodology assessed the issue of facial recognition using the Colorado State University Face Analysis and Assessment System. [6]. The FERET database was used to implement this research. The FERET dataset consists of 14051 greyscale images representing 1,199 individuals. Only images with a frontal face with variation in lighting condition, facial expression, and pose angle were considered in this research [22].

(b) Key-Point Based Techniques

The key-point based techniques are used to detect specific geometric features, according to some geometric information of the face surface (e.g the distance between the eye, the width of the head). These techniques can be defined by two significant steps, key-point detection and feature extraction.

Scale Invariant feature transform (SIFT): SIFT is an algorithm used to detect and describe the local features of an image [23]. This algorithm is widly used to link two images by their local descriptors, which contain information to define a match between them. Sift algorithm is realized in four major steps: (1) detection of maximum and minimium point in the space scale. (2) location of characteristics points, (3) assignment of orientation, and (4) descriptor of characteristics points. A framework to detect the key point based on key point was proposed by where SIFT was used in conjunction with Kepenekic approach for face recognition [24].

2.2.2 Holistic Approaches

In this approach, complete face or a single feature is considered as a target point for face detection and recognition and compared to the similarity with the whole face [25]. Holistic or a single feature approach ignores the individual features like eyes, mouth and nose. Holistic approach is further classified in to two schemes shown in figure 2.2.

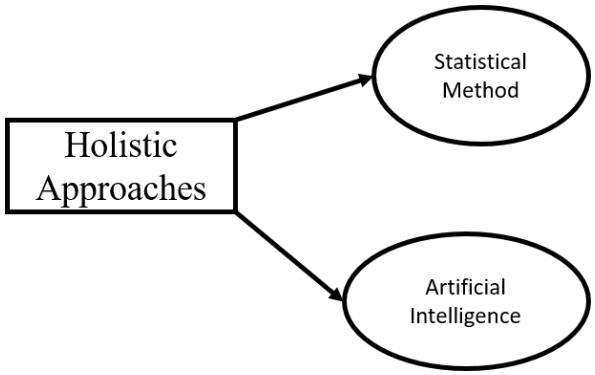


Figure 2.2: Holistic face recognition methods

a) Statistical based approach

In the statistical method, the face image density is calculated and the density set values are compared with the density of images present in the database. The calculation for this comparison is very expensive and less accurate because of orientation, scaling, and lighting. To overcome this problem, it was suggested to reduce the size of images. Scaling of images will help in comparison of images reducing the computational complexity. PCA, ICA, Fisher discriminant Analysis and probabilistic eigen faces are few methods based on statistical approach [26][27].

b) Artificial Intelligence based approach

Artificial intelligence (AI) based approaches utilizes neural network and machine learning techniques to detect and recognize faces. AI based neural network need large training sets and weights were assigned to the neurons based on the training sets. These neurons work as a filtering agent for identifying and image. Test image will pass through those neurons and the result will be with highest probability of similarity. These approaches are computational expensive but have good results [28].

Convolutional Neural Network is a relatively newer technique for face detection and has been competing with other techniques for image classification and recognition. There has been a great increase in the performance of certain

machine learning activities [19][29]. Deep convolutional neural networks in the recent past have received success in image classification and other recognition tasks [30] due to the availability of GPU graphical processing units for computing [21].

CNN-based approaches are more divided into two types a) Multi-stage detection and b) one-stage detector. The performance of both detectors has it own pros and cons. The algorithm classification is shown in figure 2.3.

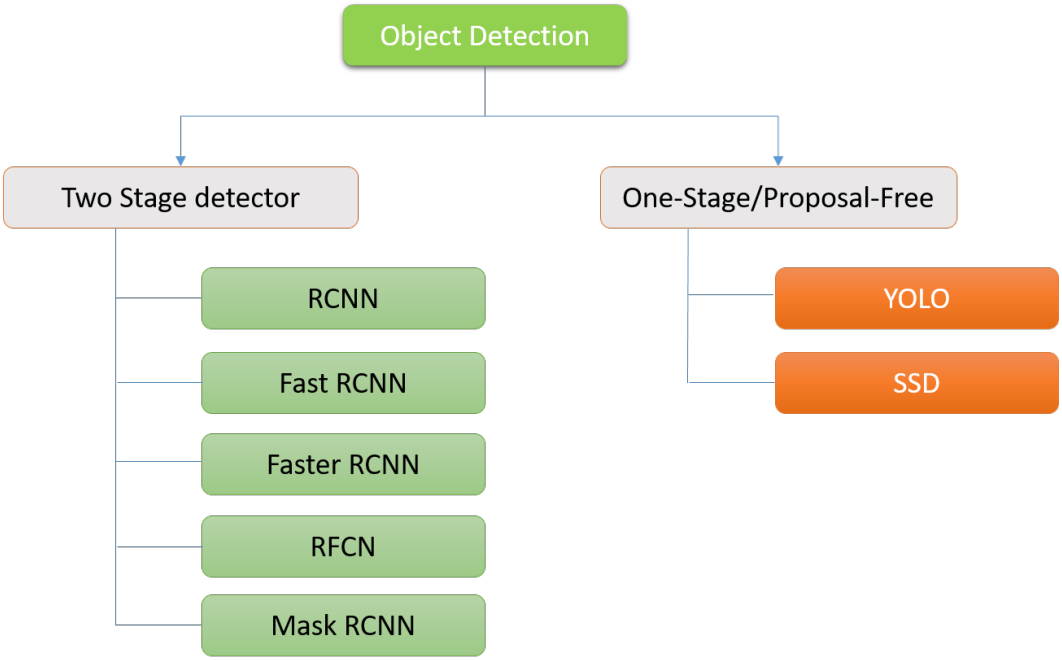


Figure 2.3: classification of Neural Networks based Object Detection techniques

The two stages of a two-stage detector can be divided by an RoI (Region of Interest) Pooling layer. One of the prominent two-stage object detectors is Faster R-CNN. It has the first stage called region proposed network (RPN), a Region Proposal Network to predict candidate bounding boxes. In the second stage, features are by RoI pooling operation from each candidate box for the following classification and bounding box regression tasks [31]. In contrast, a one-stage detector predicts bounding boxes in a single step without using region proposals. It leverages the help of a grid box and anchors to localize the region of detection in the image and constraint the shape of the object.

The accuracy of CNN based algorithm is determined by mAP. YOLO and RCNN both have good accuracy for object detection. The selection of the algorithm is dependent on the usage. As YOLO is more faster in results but RCNN has better results, so there is always a trade off between these. Figure 2.4 will describe the mAP for the most used NN algorithms.

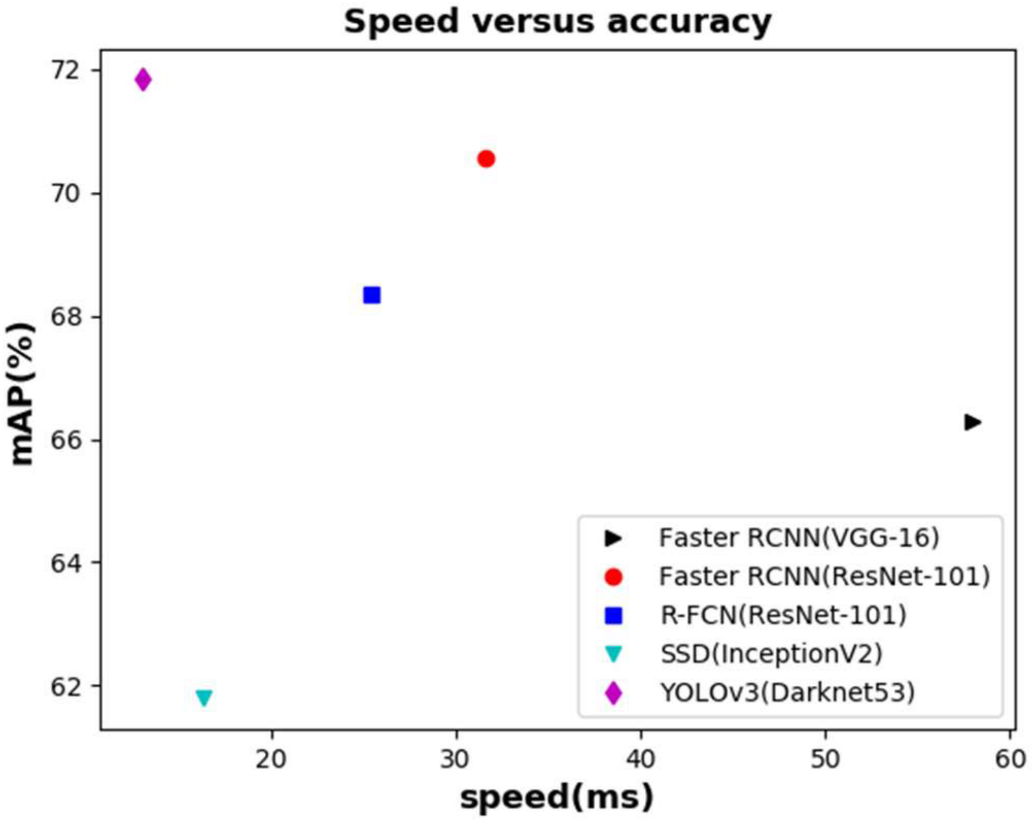


Figure 2.4: Speed (ms) versus accuracy (mAP) [5]

The model object detection is also sensitive to small object detection. IOU with different threshold was also used in this research to see the result o different objects. The result is shown in Figure 2.5.

Krizhevsky, Sutskever, and Hinton have learned a massive deep neural network to identify 1,2 million high-resolution images in 1,000 distinct groups called alexnet [32]. The said CNN had 60 million parameters and 650 thousand neurons and consisted of five convolutional layers and 3 fully connected softmax

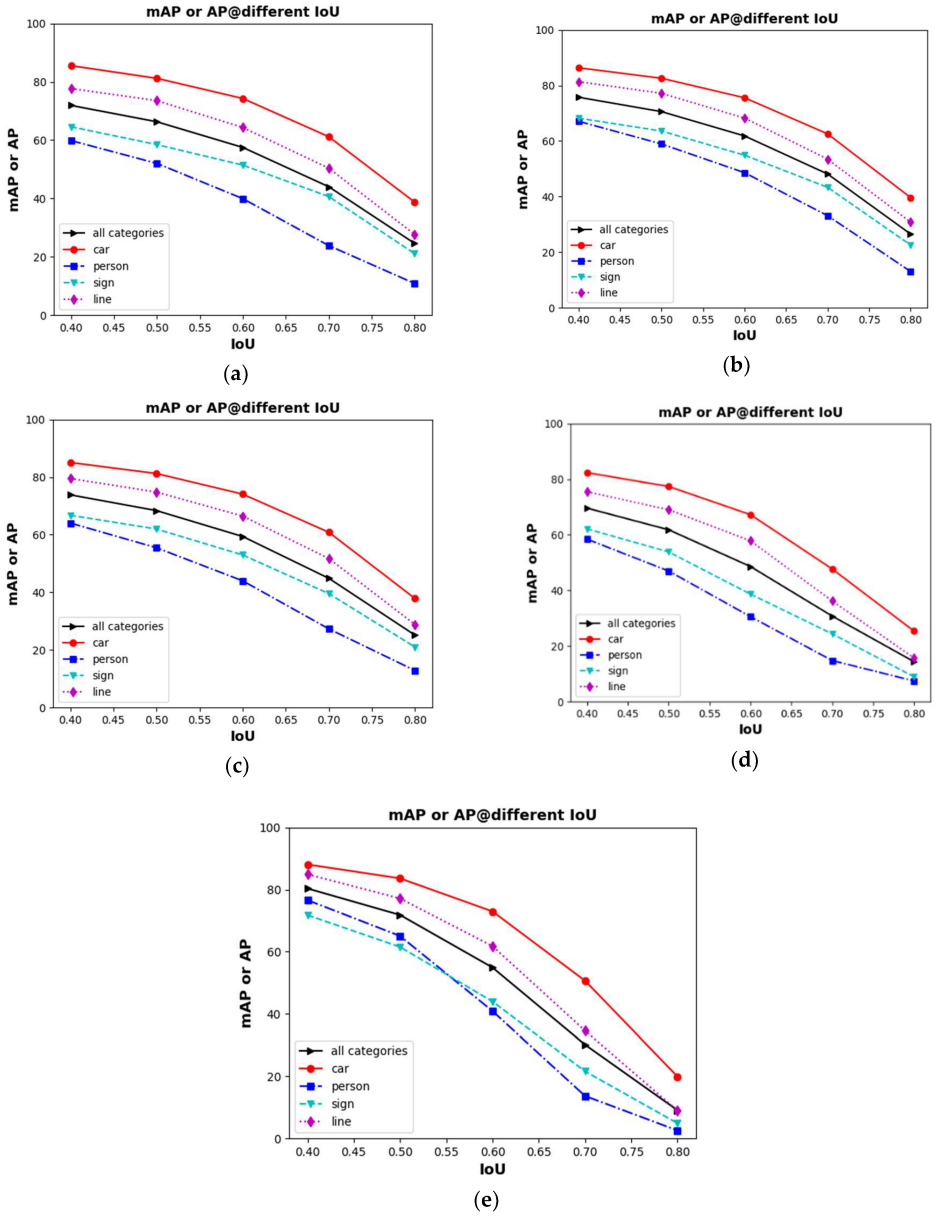


Figure 2.5: Detection performance under different IoU threshold for each detector when predicting. (a) Faster RCNN(VGG-16); (b) Faster RCNN(ResNet-101); (c) R-FCN(ResNet-101); (d) SSD(InceptionV2); (e) YOLOv3 (Darknet53) [5].

layers. This was trained over an image net data set which had 15 million labelled high resolution images belonging to roughly 22 thousand categories. This convolutional neural network obtained a top-1 and a top-5 test set error rates of 37.5% and 17.0% on the image net dataset.

The use of deep neural network has proved to be useful for image recognition and recognition, it requires a large range of training data, and thus more computing power is needed to achieve acceptable accuracy. There are many other object detection algorithm like single shot detection(SSD) and Region-Based Convolutional Neural Networks(R-CNN). All of these neural network based algorithms needs huge data set for optimal performance.

You only look once (YOLO), a type of DNN, is another new approach for object detection. YOLO performance is faster than the existing object recognition techniques. Fast YOLO processes 155 frames per second, which is double in number than the other existing approaches for video processing. Joseph Redom took 45 frames per 14 second video without batch processing on GPU and showed promising results of streaming video processing in less than 25 milliseconds latency [33][34].

Therefore, with this high accuracy and low latency, video streaming face detection and recognition is achievable in complex low lighted environment. YOLO makes far fewer background mistakes as compared to different CNN approaches. The detection time is lesser than the previously existing CNN approaches [33]. Redmon connected YOLO with web-cam to check the real-time face detection and found that YOLO process images individually and behaves like a tracking system, where it can detect objects as they move around and change in appearance [7].

The algorithms are evaluated based on average precision. Average precision (AP) is calculated with the help of intersection over the union. The AP is of YOLO is better than RCNN this will decrease the number of errors. In this research, YOLO was selected to perform face recognition.

Holistic approaches have few advantages over other techniques. The main advantage is that they don't destroy any of the information in the image by concentrated on only limited regions or points. However, this is also the biggest disadvantage for using these techniques, All the techniques lies under this method start out with the basic understanding of equal pixel importance for the whole image. This assumption not only make these techniques computationally expensive

but also requires a high degree of correlation in between test and training dataset. The accuracy of these algorithm also effected by large variance in pose and scale.

2.2.3 Hybrid Approaches

A hybrid approach in face recognition has a special status. It comes up with a combination of both geometric and appearance approaches. These two approaches combine in parallel or series to overcome the shortcomings of the individual approach [35]. Hybrid approaches give us a better result than using only one approach in solving this face recognition problem. A classical method LBPH with KNN also used a series structure for appearance based and geometric approaches. In the first step of detection, Haar cascade was used to detect the faces from the video sequence and pass to LBPH for the extraction of features which are further passed to KNN for the classification. There are other examples where Principal component analysis (PCA) and neural network were combined to achieve face detection and recognition.

2.2.4 Mostly used Face detection and Recognition Techniques

In the past few years, face recognition has been appreciated as one of the most promising applications in the field of image analysis. Face detection can consider a substyle part of face recognition operations. There are number of methods which showed good accuracy for face detection. The goal of all those algorithms is to find whether the face is present in the image or not. If the face is present in the image, then it will return image location as x and y axis.

Face detection is a segmentation problem, the image is segmented into two parts: one contain faces and the other representing non-face regions. The outcome of the algorithm is to find the face image in the frame and return the image to a rectangular box. Most famous algorithms include Viola Jones, Genetic algorithm, CNN, SVM and Hough transformation with convolutional neural network.

Viola jones object detection framework was proposed by Paul Viola and Michael Jones in 2001. Initially it was used for object detection [36]. however with time its usage expanded to human face detection, which is a sub domain of object detection. In [37] Viola-Jones consists of three techniques for the facial part detection 1) the Haar like features for feature extraction. The presence of the feature in any image is determined with Haar feature 2) Ada Boost machine-learning method for detecting the face. 3) Cascade classifier used to combine many features efficiently. Haar-Cascade is main part for face detection. The Haar-Cascade have been allowed to convolve and scan through the image from the top left corner and it ends the detection of the face processing at the right corner.

Vikram and Dr.S.Padmavathi present a system which uses Viola- Jones Cascade Object Detector for detecting face in an image and locating the facial features like eyes , nose and mouth in it. The experimentation shows 92% accuracy on Bao database with varying lighting conditions and pose variation [37].

In recent years, different architecture and models of ANN were used for face detection. Rowley, Baluja and Kanade presented face detection system based on a region based neural network. This examines a small window of an image to decide if the face is present or not. This techniques undergo multiple stages before final result. In the first step it applies the neural network based filter on image. This uses an attributor to combine the output. The filter applied calculates each location in the image on several scales to decide whether face is present in it or not, In the last step it merges the detected regions and omits the overlapped area. The result of this method is not good as Viola Jones in pose variation and this is computationally expensive for face. This technique is beneficial in object detection[38].

KNN is a traditional approach in computer vision, which is still being used for classification of objects and face recognition. It is mostly used in conjunction with a feature extraction algorithm which is generally LBPH. Abuzneid and Mahmood deployed LBPH and KNN for face detection, using the Yale Face Database B and ORL database to find the closest neighbors with more than 90% precision [39]. The recognition rate was calculated using equation (2.1) [40].

Table 2.1: Confusion matrix

		Prediction outcome		
		p	**n**	**total**
actual value	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

$$RR = \frac{N_C}{N_T} \times 100 \tag{2.1}$$

Where RR is the recognition rate and it will be measured in percentage. N_c is Number of correct matches and N_T is number of images in training set.

To analyse the algorithm and find its efficiency, a confusion matrix was created using test dataset. A confusion matrix is a table often used to describe the performance of a classification model (or classifier) for a set of test data for which true values are specified [41]. Accuracy of every algorithm was calculated by a confusion matrix. The matrix is shown in table 2.1.

While the findings of KNN with LBPH are very promising, the downside of this classical approach is high computing time with large sized images. The high computation time is attributed to the time taken to compare the test picture with the full training dataset with the complexity of O(t(n)), where t(n) is the complexity time that depends on the time used to measure the closest K distances [19].

2.2.5 Face Detection and Recognition in Low Illumination

A number of factors affect face detection and recognition, including low lighting. Lighting has a very strong effect on images, the difference of lighting on the same image provides different information [15]. Performance of even most successful face detection algorithms such as viola's work, typically fail in poor lighting or in low SNR (Signal to Noise Ratio) images [42]. Beveridge, explains multiple conjectures to describe the effect of light and focus in image processing [43]. The probability of authentication for front-light images is 0.95. The probability reduces to roughly 0.65 for images where side-lighting is more pronounced. Image lighting degrades the performance of the image recognition algorithm due to poor contrast and high degree of false-edge detection [44]. Low illumination and noise in the images plays an important role in false edge detection. Edge detection helps in reducing the the data points in data extraction process. With low illumination images false-edge detection will reduce the overall detection and recognition accuracy [45].

Abad implemented a face recognition using LBPH, OpenCV and completed processing using Fisher face and Eigenfaces algorithms [46]. The research focused on the enhancement of preprocessing of low illumination and pose variant images. Frontal faces from Yale Face Database of 20 individuals with 64 pose variants and different luminosity were used to identify the speed and identification rate. The performance of Fisher's face and LBPH were found to have the same accuracy.

2.3 Conclusion

From the literature review it may be said with confidence that both LBPH with KNN, and CNN give accurate and comparable results when face detection and recognition is done using similar image database. CNN is good in object detection and face recognition but it needs an enormous amount of data. In regard to real time detection of faces, YOLO has proved to be a step ahead of the conventional approaches. YOLO has not been extensively used for low light video-based

environment research, therefore, the potential of YOLO needs to be explored in this domain.

Chapter 3

Machine learning Techniques for Face Detection and Recognition

3.1 Introduction

Face detection and recognition is a significant biometric technique for identification. This technique has been widely used in many areas, such as military, finance, and public security [47]. Over the decades, a number of algorithms and approaches have been developed for face detection and recognition. However, the two widely used approaches are local feature-based and appearance-based approaches. The most commonly used technique is the K-nearest neighbor (KNN) which is used in conjunction with local binary pattern histogram (LBPH) [48][49][50]. KNN and LBPH is based on the appearance based approach and has always provided the most promising results for face detection and recognition. Recently CNN based techniques gained a lot of popularity in object detection. Two very promising techniques in this regards are **Alexnet** and **YOLO** primarily used for object detection.

3.2 KNN with LBPH for Face Detection and Recognition

K nearest neighbors is a classification technique which is used for object and face detection. It works in conjunction with a feature extractor. One of the commonly used feature extractor for face detection and recognition application is LBPH [51]. The goal of face detection is to detect and locate face in the image. LBPH algorithm is the combination of local binary pattern (LBP) and Histogram oriented gradient (HOG) descriptor. LBP is a simple and powerful way to extract and label the pixels in an image. It is easy to represent and face the image in a straight forward vector using LBPH.

a) Local Binary Pattern

The LBP operator is one of the widely used in computer vision applications [52]. It has been shown to be highly biased and its key advantages, namely, its consistency in monotonic-level changes and computer efficiency, makes it suitable for tasks that require image analysis. The concept of using LBP in facial expressions is inspired by the fact that the face can be seen as a combination of micro patterns that are well defined by this operator.

The LBP operator was originally designed for texture definition. The operator provides a label to all image pixels by blocking a 3x3 area of each pixel with a medium pixel value and considers the result as a binary number. After that, a histogram of the labels can be used as a texture definition. See figure 3.1 with an image of the basic LBP user.

To cope with textures at specific scales, the LBP operator has become later extended to apply neighborhoods of various sizes [53]. Defining the neighborhood as fixed sampling points calmly spaced on a circle targeted at the pixel to be labeled allows any radius and variety of sampling factors. Bi linear interpolation is used whilst a sampling point does no longer falls in the center of a pixel. In the following, the notation (P,R) may be used for pixel neighborhoods,

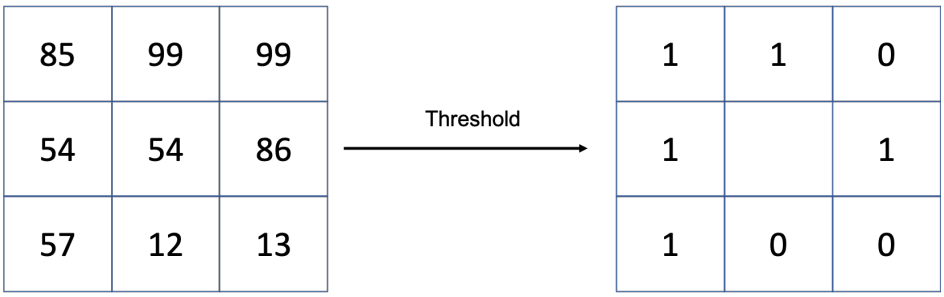


Figure 3.1: Basic LBP operator

and this means that P sampling points on a circle of radius of R. See 3.2 for an example of round neighborhoods [6].

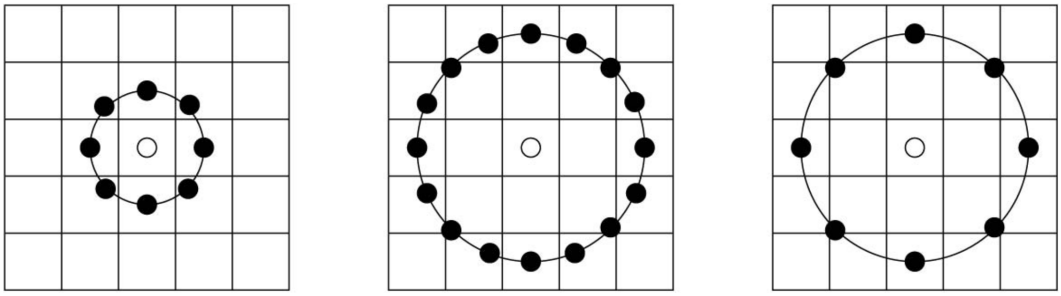


Figure 3.2: The circular (8,1), (16,2), and (8,2) neighborhoods. The pixel values are bilinearly interpolated whenever the sampling point is not in the center of a pixel[6]

LBPH is an approach which is commonly used for the extraction of features in computer vision. The idea of LBPH was obtained from local binary pattern (LBP). The LBP operator describes the contrast information of a pixel with its neighborhood pixels.The mathematical expression for LBPH is given in (5.4). LBP operator defines a 3×3 window, using the centre pixel as a reference, grey value carries out comparison with the adjacent 8 pixels [54]. These values are transformed to binary using the method that if a pixel's neighborhood value is greater than or equal to the pixel's reference value, the value is labelled with 1, otherwise 0, as seen in the equation (3.1):

$$(x) = \begin{cases} 1 \text{ x } \geq 0 \\ 0 \text{ x } < 0 \end{cases} \tag{3.1}$$

$$LBP(x) = \sum_{i=1}^{8} s(G(x^i) - G(x))2^{i-1} \tag{3.2}$$

where $G(x^i)$ is the gray value of 8 neighborhoods of the pixel G, the radius of which is R. G(x) is the gray value of the pixel.

The LBPH algorithm uses the circular LBP operator which is an improved version of LBP operator. This improvement makes the LBP operator no longer limited to the fixed radius and neighborhood. It requires significant processing time and an enormous amount of resources for the correlation method used in LBPH. In the facial recognition scheme, reduction of features and the representation of the face is also necessary [55]. In computer vision, image analysis, and pattern recognition, LBPH is typically the chosen method; it is ideal for feature extraction because it defines an image's texture and form.

The K-nearest neighbor algorithm (KNN) is a method to identify objects from the feature extractor based on the nearest training examples in the feature space. KNN is a form of instance-based modeling (lazy learning), the lazy learning function is only locally approximated, and all computation is delayed before the training data set is classified. The algorithm's training method consists only of storing the basic features of the vectors and the class labels of the training samples. K is a user-defined constant in the classification task and an unlabeled vector is classified by assigning the most common mark among the k training samples closest to that query point. There are three distance equations for K-nearest neighbors for grouping, i.e. Euclidean distance, distance from Manhattan, and distance from Minkowski. The most widely used distance metric is Euclidean distance for a constant variable. Equation (3.3) shows the Euclidean expression.

$$d(a, b) = \sqrt{\sum_{i=1}^{m} (a_i - b_i)^2} \tag{3.3}$$

Where the difference between two variables' values is taken and squared and summed for points. a and b are two points and the value and m will 2 in this case.

The Yale Face Database and ORL is the dataset commonly used by researchers for KNN and LBPH preparation. The Yale Face Database is a regular database containing 165 GIF-format grey scale photographs of 15 individuals [56]. Every subject consists of 11 photos. Center-lighting, with or without spectacles, satisfied, left lighting, right light, and the wink are the various setups [57]. The ORL has 10 different images for each of the 40 distinct subjects [58]. The photographs are taken at various times for certain subjects, with variable light, facial expressions (open or closed eyes, smiling or not smiling) and facial features (with or without spectacles). With the subjects in an upright, frontal position, all photographs are shot against a black, homogeneous backdrop.

Table 3.1: LBP+KNN Recognition accuracy with different distance method

LBP+KNN recognition Rate		
Method	Yale Face Database	ORL
Euclidean Distance	91.1 %	93%
Manhattan Distance	91%	94%
Correlation Distance	90%	94%

KNN suffers from the problem of dimensionality or overfitting, if dimensions of the features increases. The only solution to avoid overfitting is to increase the data regularization, but it increases exponentially as the dimension increases. Similarly, in KNN, as the number of features increase, the size of the data is also required to be increased to achieve the desired accuracy, but then it also becomes computationally more expensive and time consuming.

3.3 Convolutional Neural Networks for Face Detection and Recognition

Deep convolutional neural networks shows great advantages in computer vision tasks. However, the network have complex network structure which exists of large number of layers such as convolutional layer and pooling layers [59]. This needs valuable memory and computational resource and needs large training time. In order to work with a small number of samples, conventional machine learning approaches often use shallow constructs [60]. The shallow convolutional neural network (SCNNB) network has only 4 layers with small size of convolution kernels, which requires low time complexity and space complexity [61]. The efficiency and generalization potential of complex classification problems emerge when the target object has a rich context.

CNN is a relatively recent face detection technique which was first used in 2012 by Alex net as an object detection technique when he used the CNN technique for object detection in the Image Net competition and won the competition [32]. CNN has been competing with other visual classification and recognition methods since then [62]. In the field of image analysis, it has been used extensively since it is excellent at dealing with problems of image detection and identification. The precision of many machine learning tasks has improved tremendously. It has a clear and universal paradigm of deep learning [63]. Deep convolutional neural networks have in the recent past earned success in image detection and other recognition tasks owing to the availability of graphical processing units (GPU) for computation [64][65].

A CNN consists of multiple types of layers. Between the input and output layers, several hidden layers are present. The hidden layers are typically sequenced as a convolutional layer, pooling layer (for dimension reduction), and fully connected layers (for connected neurons of layers). In the shape of a vector, the input layer accepts the input and transforms it across the hidden layers. Every hidden layer is made up of neurons; these neurons are not connected to any of the neuron within a layer, but fully connected to all neurons in the previous

layer. The convolutional hidden layers convolve with a multiplication or other dot product activated through a Relu layer, followed by additional convolution through the pooling and fully connected layers. The output layer of the neural network that describes the class scores is the last fully connected layer. The solution will work from the perspective of image processing as given below:

- **Input Layer:** The input layer is the first layer in CNN and it comprises image data. Image data generally consist of pixel values, which were represented in the form of a matrix. to feed this to input layer, it is essential to convert the image data in to single column. For example, an image with dimension 32×32 will be converted in 1024 before feeding to the input layer. This single column contains the width, height, and color detail for that image. Figure 3.3 elaborates the working of input layer.

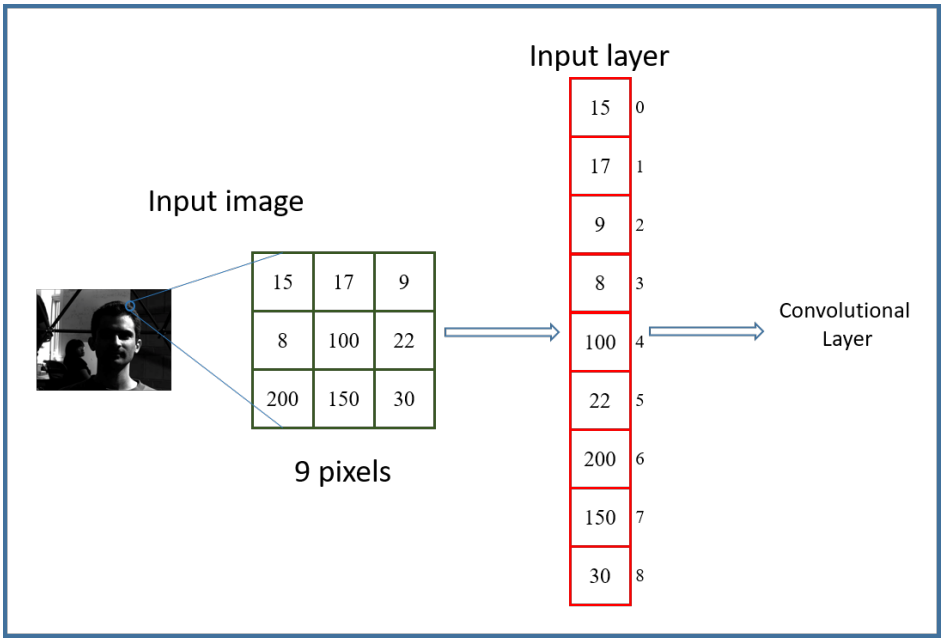


Figure 3.3: Workflow of Input layer in Convolutional Neural Network

- **Convolution layer:** The convolutional layer is called the feature extraction layer. In this layer features from the input image are extracted. In the first step, the Convolutional layer performs a convolutional operation on a part of an image. Convolutional operation calculates the dot product between the

convolutional kernel (filter) and the receptive field (local region of an image which has the same size as the kernel). It results in a single integer as output volume. The kernel will slide over every receptive field and do the same operation on every iteration. This operation will be repeated multiple times to go through complete image. This output will be an input feed for the next layer. Figure 3.4 describes the working on convolutional layers.

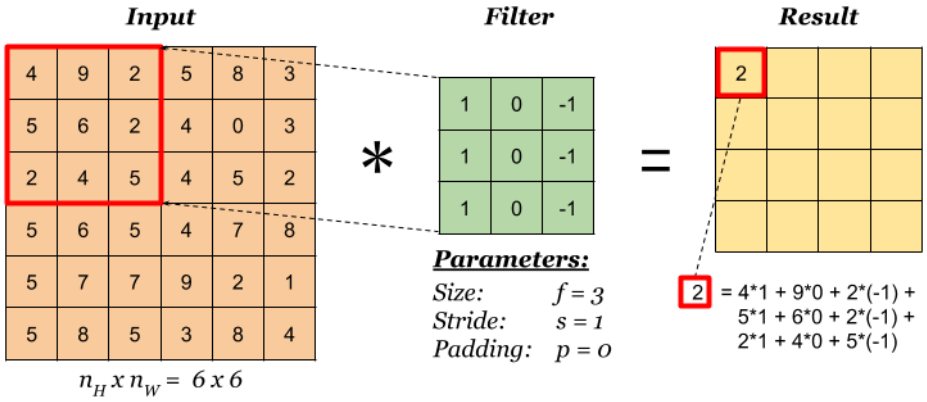


Figure 3.4: Structure of Convolutional Layer

- **Rectified Linear Unit(ReLU) layer:** The most widely used activation function in neural networks is the Rectified Linear Unit (ReLU). In conjunction with the convolutional layer, ReLU activation works. It depends on the values obtained from the convolutionary layer. The function returns 0 if any negative feedback is obtained, but returns the same value for any positive value. So this layer helps to omit from the neural network the false value bias effect in this sense. Any time a convolution layer is added to the ReLU image layer, the non linearity of its values is reduced.

- **Pooling layer:** The Pooling layer will be next to the convolution and ReLU layer in the neural network structure. Every layer in the neural network, every layer feeds the next layer to its output. The pooling layer is used to reduce the spatial dimension of the input image. This dimension reduction technique helps in reducing the computation in the completion of the process. There are different types of pooling layer which includes max Pooling, average pooling, and mean pooling.

Max pooling was also used as a noise suppressant. It also helps in discarding the noise from the image along with dimensional reduction. The performance of max pooling is better than average pooling. Figure 3.5 will explain the pooling.

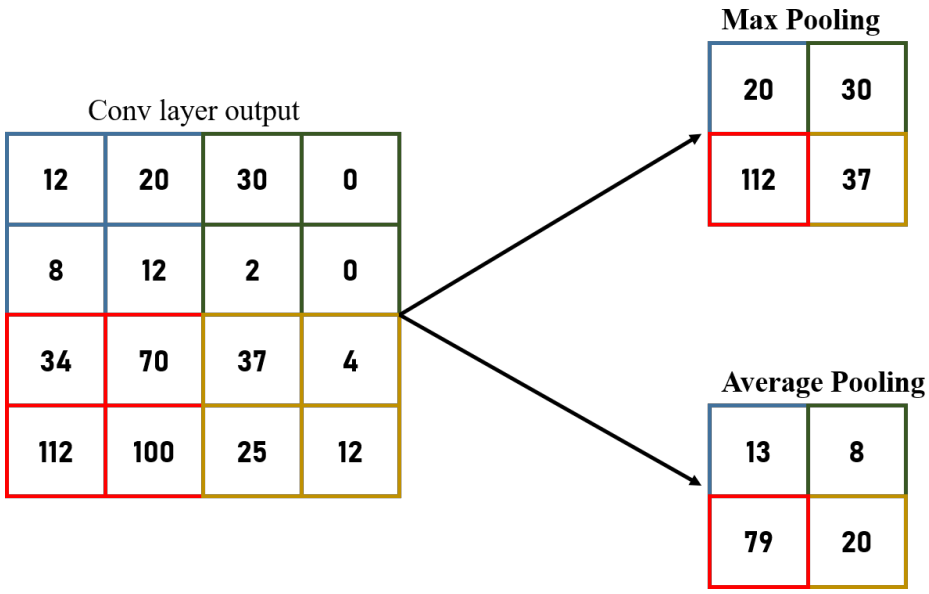


Figure 3.5: Schematic representation of max and average Pooling with filter 2×2 with stride 2

- Fully-connected layer will compute the class scores.

The technique has been successfully implemented for face detection and recognition from video sequences using five layer CNN and a registered database [66]. The detection rate was calculated using equation (3.4) and (3.5).

$$DetectionRate = \frac{TP}{TP + FN} \tag{3.4}$$

$$AccuracyRate = \frac{TP + TN}{TF} \tag{3.5}$$

Figure 3.6 will describe the basic structure of Convolutional neural network:

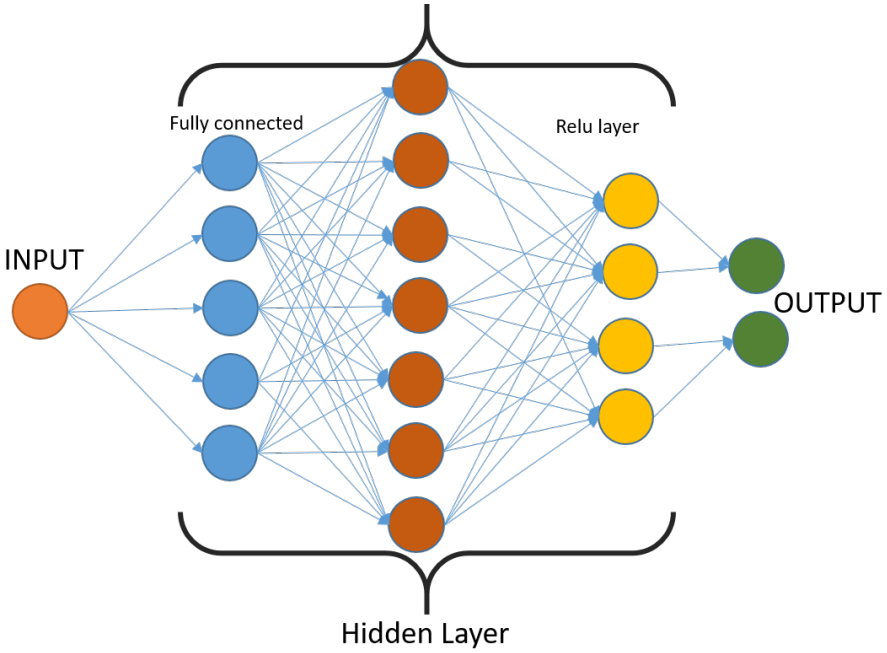


Figure 3.6: Convolutional neural network structure

- TP (True Positive): The frame in which the bounding box exactly detects the face is the value of TP.
- TF (Total Frames): The total number of frames present in a video sequence.
- FN (False Negative): It is the number of frames when the bounding box does not detect any face or the face does not fall in the bounding box.
- FP (False Positive): The number of frames where no object is present and the bounding box does not fall within the bounding box.

3.4 You Only Look Once(YOLO)

You only look once (YOLO) is a region-based convolutional neural network(R-CNN) that has effectively been used for object detection for fewer background mistakes and at considerably higher frames per second making it more suitable for image recognition from video streaming. The results from YOLO, RCNN and Fast

RCNN [7] are shown in Table 3.2 which shows the time taken by four different CNN based face detection approaches with YOLO clearly standing out as the fastest approach.

The critical difference between YOLO and basic CNN approaches is that CNN approaches take more time to classify the image because the feature detector kernel checks every pixel of the image every time to detect the required feature. However, YOLO first runs the selective search and creates a bounding box around the region of interest (ROI). Afterwards, the ROI feeds into any network for the classification. This complete process reduces the computation required for applying a number of feature based kernels to the whole image [67].

YOLO is a detection technique for objects. It frames object detection into spatially segregated bounding boxes and related class probabilities as a regression problem. In one evaluation, a single neural network predicts bounding boxes and class probabilities directly from complete images. YOLO divides the input image into a pixel size grid of $S \times S$. If the center of the object lies in a grid cell, the grid cell would be responsible for detecting the object. Each grid cell predicts B bounding boxes and a confidence score for these boxes. The confidence score reflects how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. The confidence score is:

$$C = Pr(\text{ Object }) * \text{IOU}_{\text{pred}}^{\text{truth}} \tag{3.6}$$

where C is confidence score, the IOU(Intersection over union) represents a fraction between 0 and 1.

The intersection is the overlapping area between the predicted box and ground truth. Ideally IOU should be close to 1, which will indicate that predicted boxes is close to ground truth.

3.4.1 Network Design of YOLO

YOLO has 24 convolutional layers, followed by 2 layers. To reduce the depth of the map features, any convolution layers use 1×1 reduction layers as an alternative. It produces a tensor in the form (7, 7, 1024) for the last convolution sheet. The tensor is then flattened. Using 2 connected layers as a method of linear regression, it produces 7×7×30 parameters and then reshapes it to (7, 7, 30), these are 2 boundary box predictions per location.

(a) Loss Function

YOLO predicts several bounding boxes for each grid cell. Each of the boxes is responsible for the object and calculate the loss using the true positive. For this purpose, it is essential to select the highest IOU (intersection over union) with the ground truth. This strategy leads to specialization among the bounding box predictions. Each prediction gets better at predicting certain sizes and aspect ratios.

YOLO uses sum-squared error between the predictions. The loss function composes of:

- Classification loss: If an object is detected, the classification loss at each cell is the squared error of the class conditional probabilities for each class:

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \tag{3.7}$$

- Localization loss: The localization loss measures the errors in the predicted boundary box locations and sizes, only the box responsible for detecting the object is counted.

$$\begin{aligned} &\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2\right] \\ &+\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} 1_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2\right] \end{aligned} \tag{3.8}$$

- Confidence loss: If an object is detected in the box, the confidence loss (measuring the abjectness of the box) is:

$$\sum_{i=0}^{S^2} 1_i^{\text{obj}} \sum_{c \in \text{ classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \tag{3.9}$$

1_i^{obj} is 1 if an object appears in cell otherwise 0.

$p_i(c)$ denotes the conditional class probability of class c in cell i.

The main issue with YOLO is that it has smaller object prediction error than other CNN approaches. YOLO selects the object by applying anchors to the frame. Simple YOLO uses five anchors to detect the image. Anchor boxes are responsible for predicting the object in the image. Small objects in the frame show the detection error with YOLO.

The matrics used in the research to quantify and evaluate the classification and detection results are:

- Precision: In classification, precision represent the number of selected relevant items. It is computed by dividing the number of true positive predictions by the sum of true positive and false predictions. The following equation is used to find precision (3.10) .

$$Precision = \frac{TP}{TP + FP} \tag{3.10}$$

- Recall: Represent the number of relevant items selected and it can be computed by dividing the number of true positive prediction by the sum of true positive and false negative predictions. Equation (3.11) is used to find Recall.

$$Recall = \frac{TP}{TP + FN} \tag{3.11}$$

- F score: It quantifies the system performance and can be calculated by using equation (3.12).

$$FScore = F_1 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.12}$$

Face detection with YOLO achieves a high degree of accuracy. Because YOLO algorithm works using the following three techniques:

- Residual blocks
- Bounding Box regression
- Intersection Over Union (IOU)

The complete frame was divided into grids and bounding box will be drawn over it with in the grid with the confidence score. last step in intersection over union(IOU), this phenomena describes how boxes overlap. As each grid cell is responsible for predicting the bounding boxes and their confidence score. The IOU equal to 1 if the predicted bounding box is same as the real box. this eliminate the bounding box that are not equal to real box. Logically these step will leads to less background error.

YOLO makes fewer mistakes as compared to different CNN approaches [68]. The detection time is lesser than the previous existing CNN approaches. Table 3.2 shown in the time taken by four CNN based approaches [69].

Table 3.2: CPU and GPU time used by four CNN approaches

CPU and GPU time used by four CNN approaches			
Method	CPU Time(s)	GPU Time(s)	Total Time(s)
Fast R-CNN	2.7	10.1	12.8
Faster R-CNN	0.3	0.4	0.7
YOLO	0	0.04	0.04
DeepIR	0.3	0.4	0.7

3.5 Conclusion

The machine learning algorithm performs well in illuminated still images. LBPH with KNN is represented as a classical approach for face detection and recognition, and many researchers implemented LBPH with KNN using different distances such as Euclidean, Manhattan Distance, and Correlation Distance. Convolutional neural networks are new technique which has a multiple layer structure, and it performs well with standard lighting still images. The low illumination or low contrast images are a significant problem in face detection and recognition. The next chapter explains the illumination and how it affects the face detection and recognition.

Chapter 4

Low illumination Dataset Problem in Face detection and Recognition

4.1 Introduction

The appearance of the object is affected by illumination in various ways. The resulting variation in lighting conditions causes difficulty in designing an image-based recognition system. To comprehend low illumination, it is necessary to understand the uttering factors for low illumination is essential. This chapter includes the discussion on illumination, the effect of low illumination on detection and the available datasets for face detection and recognition.

4.2 What is Illumination?

Luminance refers to the calculation of the amount of light produced by a solid angled surface, flowing through or reflecting from it [70]. It also suggests how the human eye can experience abundant luminous intensities. This suggests that the intensity of light produced or reflected off a surface is indicated by luminance.

Illuminance is a concept that explains the calculation over a given surface region of the volume of light falling on (illuminating) and scattering.

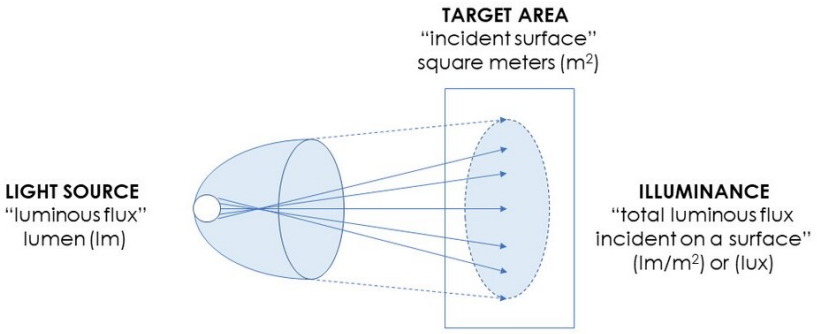


Figure 4.1: Measurement of light

Illuminance also shows a relationship with how persons perceive the brightness of an illuminated area. Illumination value is quantified using the lux meter. Light level or illuminanace is the total luminious flux incident on a surface per unit area.

Luminance can be expressed by equation (4.1):

$$E = \Phi/A \tag{4.1}$$

where E is Illuminance (lm/m2, LUX), Φ is the luminous flux. The quantity of light emitted by a light source (lumen, lm), and A is area in m^2 Illumination in a space can be calculated from equation (4.2)

$$I = L_I C_u L_{LF}/A_I \tag{4.2}$$

where, I = illumination (lux, lumen/m^2)

$L_l = luminanceperlamp(lumen)$

C_u =coefficient of utilization

L_{LF} =light loss factor

Al = area per lamp (m^2)

Normal illumination standard is 100 to 210 lux for a good representation of image [71]. Low illumination typically refers to 20 to 70 lux. There are different standards for lighting in the mining industry around the world. Standards for some of the countries are shown in table 4.1 [72]:

Table 4.1: Country Based Standards for Illumination in workplace

Country standard for Illumination in workplace [72].			
Country Name	Hualages	Workshops	Offices
South Africa	20	400	20
Australia	20	100	100
United Kingdom	70	50-100	8
United States	-	-	15
Germany	20-160	400	016

A number of factors can confuse the detection and recognition of an object(face), out of which illumination is an important factor. Illumination can make two images of the same person less similar [63]. Beveridge developed multiple models with varying light and focus to find the illumination effect using the following conjectures [43].

- Conjecture 1: Lighting subsumes the setting as a predictor in performance (Indoor vs Outdoor).
- Conjecture 2: Lighting subsumes edge density as a predictor of performance.
- Conjecture 3: Focus explains recognition failure better.
- Conjecture 4: Since focus is distinct from lighting, predictions about recognition than edge density.

The results of Beveridge show that face recognition from uncontrolled images is a challenge along with high edge density in images predict failure [43]. He also found that light intensity and side lighting strongly degrades the performance of an algorithm.

4.3 Datasets for Face Detection and Recognition

Face detection and recognition is highly dependent on dataset. The performance and accuracy of algorithms are also dependent on the training and test set of the dataset. There are several currently existing datasets. Most of these datasets are open source which are available for the research purposes. Literature review explains the used dataset for research. The mostly used datasets for the kind of research being undertaken are color FERET Database, AR dataset, Yale Face dataset, and extended Yale Face Database B. These datasets consist of images with different face poses and illumination variants.

4.3.1 FERET Dataset

The FERET dataset establishes a large dataset for facial images in an uncontrolled indoor environment using several surveillance cameras.The database collection was a collaborative effort between Dr. Wechsler and Dr. Phillips. The database contains 1564 sets of images with a total of 14,126 images. This dataset can be used for training and testing for time related face algorithms because this dataset was collected from 1993 and 1997 with the same subjects. A duplicate set is a second set of images of a person already in the database and was usually taken on a different day. For some individuals, over two years had elapsed between their first and last sittings, with some subjects being photographed multiple times. This time lapse was important because it enabled researchers to study, for the first time, changes in a subject's appearance that occur over a year. There is also version 2 available for this dataset with color images [73]. The samples are shown in figure 4.2

4.3.2 ORL Database of Faces

The ORL database contains 40 distinct object with ten different images of a single subject. It consists of grey scale images with different lighting conditions, facial expressions, and facial details. The image files are stored as PGM format with



Figure 4.2: Part of the face images from the Feret database

the size of each image is 91×112, 8 bits, and gray levels [58]. Fig. 4.3 shows example for 10 individuals, 10 images per personal from the ORL database for human face images using in training and testing [74].

4.3.3 Extended Yale Face Database B

The Yale Face Database B focuses on lighting. It comprises of 5760 single source photographs for 10 subjects. Subjects are seen in 9 poses and 64 lighting conditions, with a total of 576 images per person. Ambient light for each subject in a specific pose has also been captured. The total number of images, including ambient illumination, is 5850 [10]. Figure 4.4 shows 5 subsets of the extended Yale face dataset B dataset. This also shows different illumination present in extended Yale face dataset B [75].



Figure 4.3: The ORL database for training and testing

4.4 Conclusion

Accurate face detection and recognition in low illumination environment solves various problems in the South African industry. It may also be very useful for similar challenges faced elsewhere, particularly by police and smart city operators. But for such an application to be effective, it has to be real time and accurate in conditions with complex background and low luminosity. The current research aims to find and develop a solution for this challenging problem.



Figure 4.4: Five subset of Extended YaleB dataset

Chapter 5

Research Methodology

5.1 Introduction

The methodology to observe the performance of machine learning algorithms for face detection and recognition in environments with low light and contrast such as underground mining environment will be discussed in this chapter. To observe and evaluate the face recognition algorithms in such an environment, it is important to understand the dictating factors such as speed of detection and recognition, complex background, object distance, and light conditions. Complex backgrounds, object distance and light condition can affect the performance of the system and reduce its efficiency.

5.2 Methodology for Proposed approaches

The efficiency of deep facial recognition models depends on multiple covariates that can be clustered or categorized into multiple categories. To assess the robustness of a deep learning convolutional neural network with respect to the defined covariates, advance deep learning model will be considered. The output function of each model as the image descriptor for the specified input face image and video series will be evaluated. Figure 5.1 illustrates the methodology of the research followed by explanation in the ensuing paragraph.

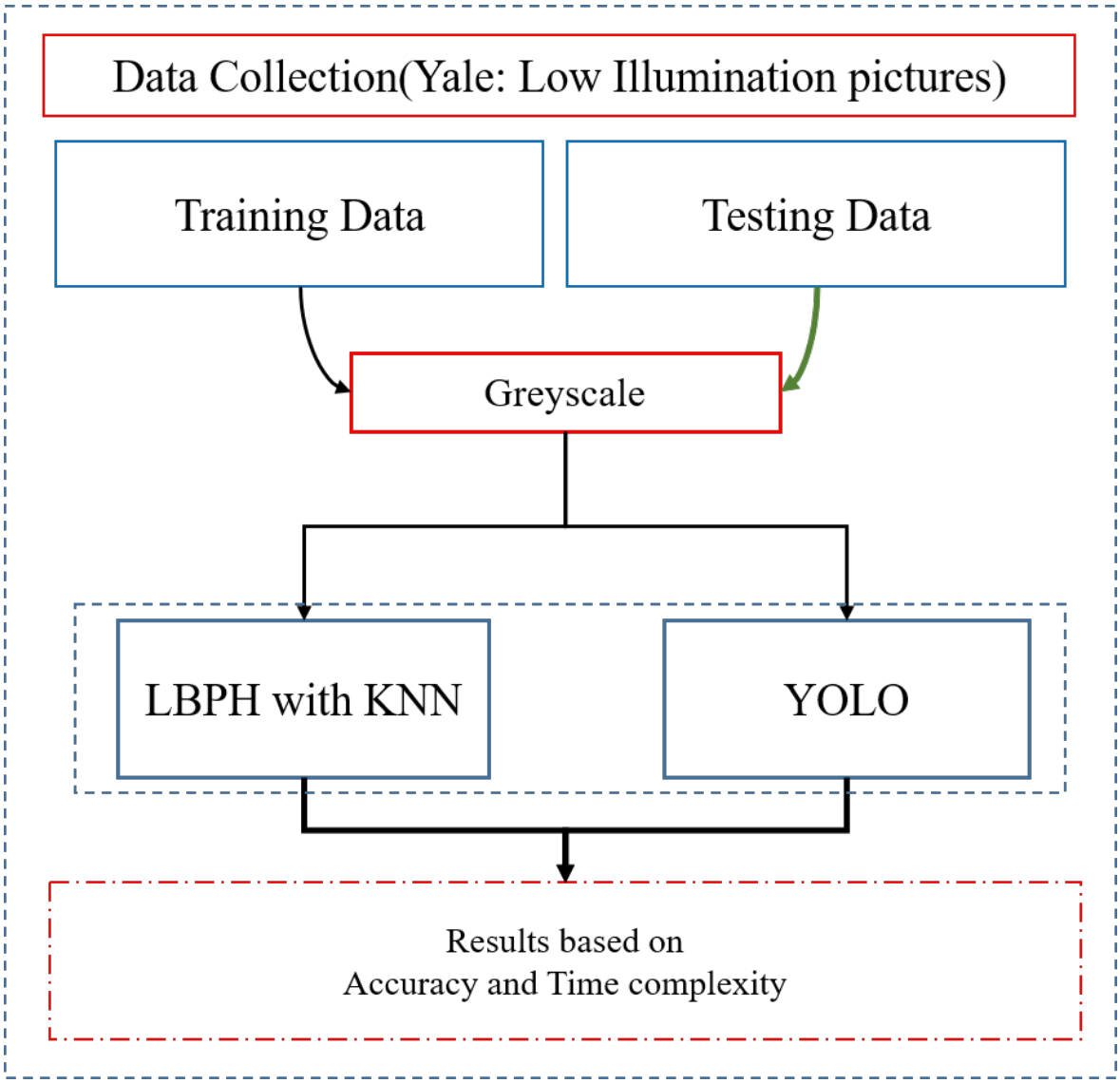


Figure 5.1: Research methodology

The LBPH- KNN approach will be used as benchmark [39]. To obtain better results from the LBPH and KNN based approach, Euclidean distance will be used to find the distance between the testing and training images. Figure 5.2 shows how LBPH helps in reducing values.

The proposed YOLO-based approach will use preprocessed images. Fig 5.1 shows the flow of the algorithm. Haar cascade detection will be used to eliminate the background effects. A coloured image has high data and their conversion

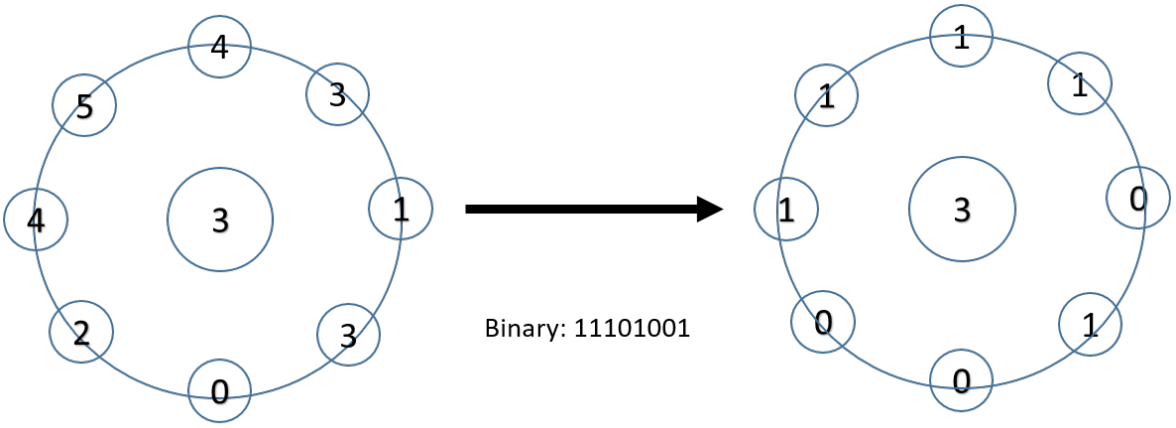


Figure 5.2: Local binary Pattern Histogram (LBPH) calculation for feature extraction

to grey-scale is expected to help in increasing the detection and recognition by reducing the dimension of the images.

The table 5.1 explains the mean average precision and frame per second rate of the different algorithms. The experimentation was done on PASCAL VOC 2007 dataset. Faster R-CNN has a better accuracy than any of the algorithms, but the limitation is the real-time usage of this algorithm. Comparison in these algorithm helps in selection of algorithm for the research. Fig 5.3 explains the error analysis of faster R-CNN vs YOLO. Using Yolo will get a real time result for face recognition as Yolo will be trained on subject faces only instead of classes of different objects.

Table 5.1: Performance comparison of YOLO and R-CNN models

Performance comparison of Yolo and R-CNN models			
Object detection Algorithm	Train	mAP	FPS
Yolo	2007+2012	63	45
R-CNN	2007+2012	70	0.5
Fast R-CNN	2007	66	7

Combining different configurations of the better variant of Fast R-CNN. Other versions of Fast R-CNN offer only a slight gain, while YOLO offers a substantial improvement in performance of object detection[76].

The frame per second rate of the algorithms defined in the table 5.1 plays an important in real time face recognition. Yolo, R-CNN and Fast R-CNN has 45, 0.5 and 7 FPS respectively. Although the mAP is lower in Yolo, but this is best suited choice for real time face recognition system.

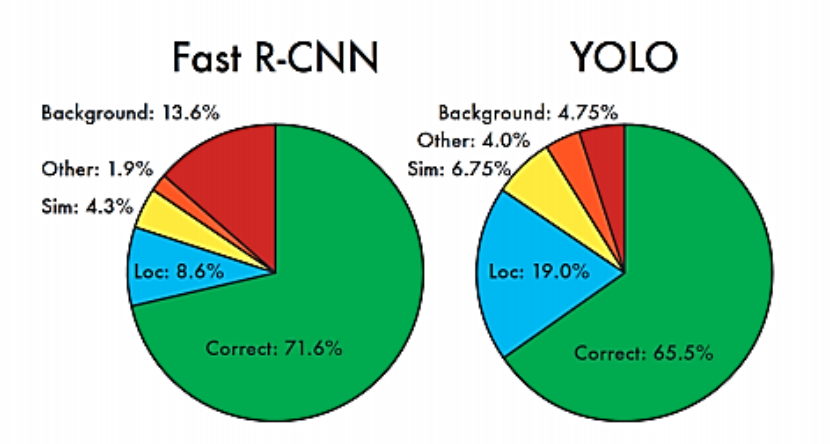


Figure 5.3: Ratio of samples and recognition[7][8][9]

CNN is a category of neural networks which are very effective in computer vision areas like face detection or recognition. CNN is a forward feed neural network made up of many types of layers. CNN is made up of neurons that have learnable weights or parameters and biases [69]. Every filter takes some input, performs a convolution and follows it with non linearity. In this research, after elimination of the background of the images using Haar-cascade, the images will be fed to the neural network (DNN) for training. The extraction of faces will decrease the size of overall images. This extraction of ROI from the complete image reduces the number of computations, which reduces the time complexity of the algorithm in real time testing [68].

CNN based approaches are well known for high precision, but CNN is a complex algorithm which takes time to compute [77]. Since R-CNN can be used to reduce the computation time in the object detection algorithm,it will be therefore used in the proposed technique. To train the algorithm for face detection in the scene, one class will be used and the image will first be divided into 3x3 boxes

to find the object. Figure 5.4 defines how R-CNN selective search works.

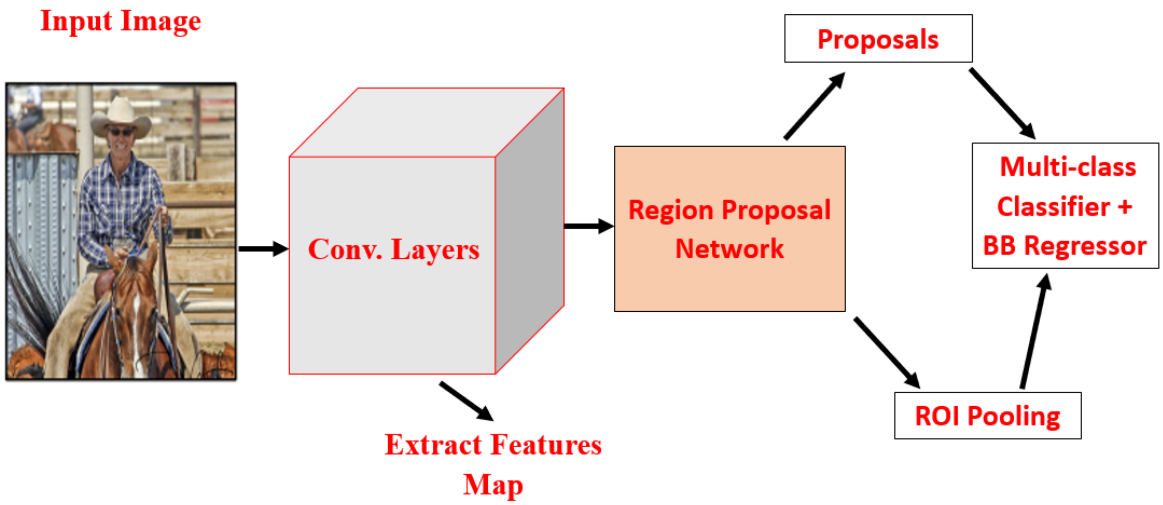


Figure 5.4: Region based Neural network Illustration

The statistic for the R-CNN algorithm is illustrative. Description: The device recognition overview. Our system captures an input image, extracts around 2000 bottom-up region proposals, uses a broad convolution neural network (CNN) to compute features for each proposal, and then classifies each region using a class-specific linear SVM [31] (Image obtained [31] [78])

The problem arises when two grid cells detect the same object. Every object has one mid point so there will be only one bounding box. However, the detection of an object from multiple grids creates multiple bounding box on a singular object. To resolve this issue, the proposed method used is Non-Max suppression.

Non-Max suppression (NMS) is used to ensure that a single object is recognized only once when an object is observed. Consider a 100×100 image with a 9×9 grid and there is a car that we want to detect. If this car is in several grid cells, NMS ensures that the optimum cell is found among all candidates where this car is located. The way NMS works is:

- It first discards all those cells where the probability of the object being

present (calculated in the final softmax layer) is less than 0.6.

$$S\left(y_i\right) = \frac{e^{y_i}}{\sum_i e^{y_j}} \tag{5.1}$$

- Then it takes the cell with the largest probability among candidates for object as a prediction.
- Finally, the remaining cells, if any with an intersection over union value = 0.5 with the prediction cell, are discarded.

YOLO is an object detector, whose anchor boxes are responsible for the detection of the object. The number of anchor boxes and the configuration decide how small an object it can detect. So to detect particular objects such as the face in face recognition, there are two possible solutions.

- Up-scaling of the image.
- Change the size and increase the number of bounding boxes.

In face recognition from a video sequence, it is challenging to upscale the image because of significant variations in face size and shape. Selecting the appropriate size and increasing the number of bounding boxes is the right approach. In YOLO for 13x13 grid cells YOLO creates 845 bounding boxes for the single image:

$$B^t = S \times S \times B \tag{5.2}$$

where B^t is the total bounding boxes, S is the size of grid and B is the bounding box in each grid.

The model for face detection and recognition depends on how accurately faces are detected from the video frames. Faces are considered as small objects in the video feed and multiple bounding boxes describe a face in face detection. To avoid multiple bounding box object detection, Non-max suppression will be implemented on the predicted boxes.

These following steps represent the implementation of Non Max suppression for face detection bounding boxes.

- Discard all boxes with where Pc $\leq$ 0.6.

$$S\left(y_i\right) = \frac{e^{y_i}}{\sum_i e^{y_j}} \tag{5.3}$$

  - Pick the box with the highest prediction value Pc.

- Discard any remaining boxes where IOU $\leq$ 0.5.

Training will be faster because of the grey-scale (reduced dimension of image). Yolo is object detection algorithm to train this for face recognition, the algorithm was trained with different subset of faces. In testing phase the input will be a video. The result will be based on class based prediction. Fig 5.5 defines the YOLO based methodology for this research.

5.2.1 Significance of Data for Machine Learning

Data sets in face recognition are very important to obtain the optimal performance of an algorithm. For this research, the Yale Face Database B obtained from an online source was used firstly for training purposes and afterward to check the algorithm performance in a real time video environment [79]. The archive includes 16128 single light source photographs of 28 subjects each viewed under 576 viewing conditions (9 poses x 64 light conditions). A picture of ambient (background) lighting has also been recorded for each subject in a specific pose.

It is a standard study procedure that 80 percent of the images of each subject are used for training, while the remaining 20 percent are used for testing, and the number of subjects in the training set in both tests are equal to the number of subjects in the test set [80]. The extended Yale B dataset has 28 subjects with 9 pose and 64 illumination variations of single object. To evaluate the model, images with small variation will be used for testing purposes.

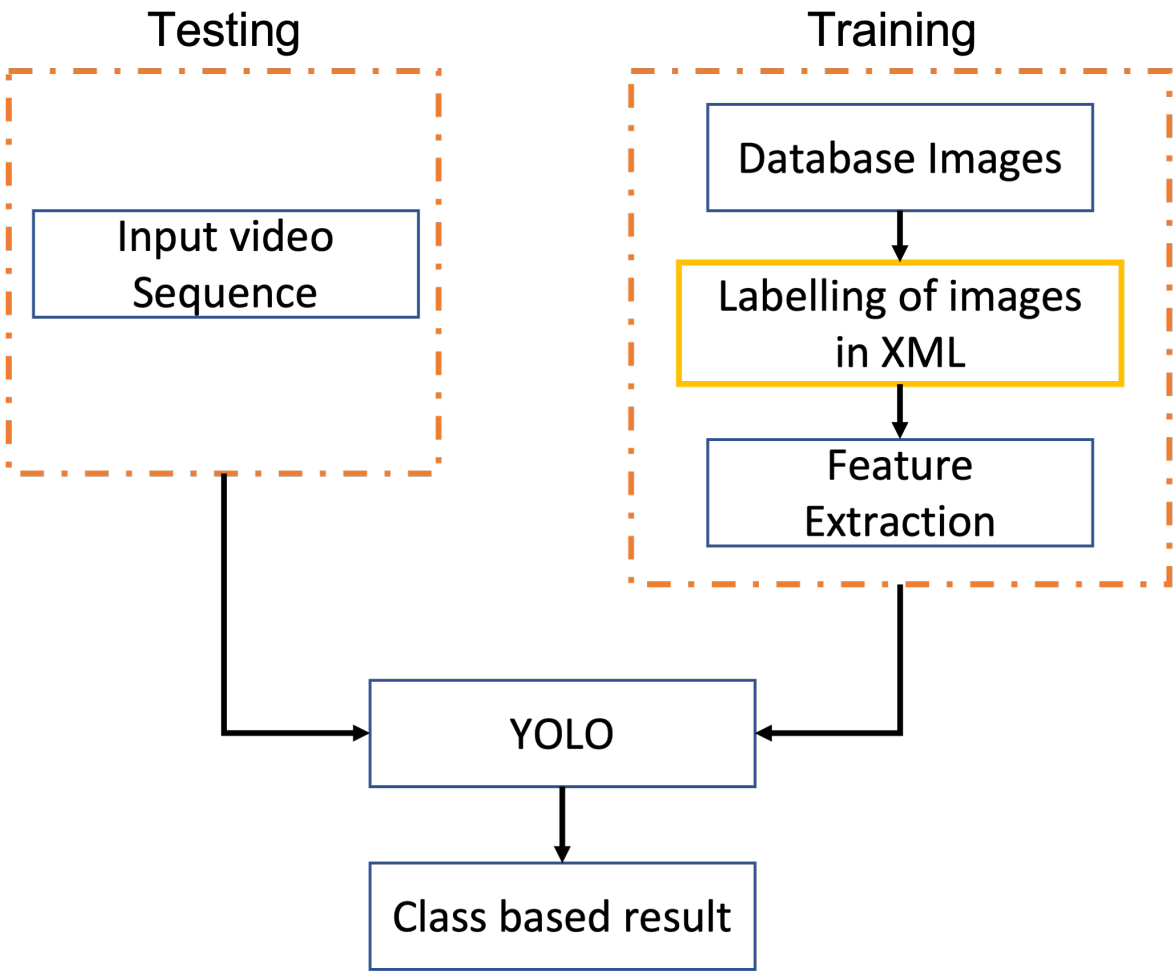


Figure 5.5: Convolutional neural network based approach

5.2.2 Implementation of models

An important part of implementing a model is to select the parameter that gives the optimal results. For this research, two models will be implemented and every model with different parameters. The first model, LBPH with KNN, has already been implemented by using parameter a value of K=6. Results are given in section 7.3. K-nearest neighbor(KNN) is a type of instance-based learning and data classification method that can be used as a face recognition method. Each pixel in the face represents some unique information [81]. A face can be recognized by using a statistical comparison of data from the training set and the test set. The image is recognized through the closest comparison of the image from the training

set. Under this approach, an image in the test set is identified by assigning the nearest point in the learning set to a label. To evaluate the proximity of the data point in the KNN, the Euclidean distance metric was used. All pixels were given a distance. To calculate the Euclidean distance between the corresponding elements of the two vector spaces (3.3) can be used.

It is a difficult and time-consuming procedure, for instance learning to compare and generate the results. The problem with this technique is that the position of faces in an unconstrained environment is different. Many algorithms have been proposed to solve this problem [10], but among all algorithms such as LBPH has generally delivered the most promising results.

LBPH is a type of visual descriptor used for classification in computer vision. It is a particular case of Texture Spectrum model proposed in 1990. It was first described in 1994 by Ojaha and its operator works on eight neighbor pixels [82][65]. The mathematical equation (5.4) of LBPH operator is:

$$d(a,b) = \sum_{i=1}^{r=8} s(G(x^i) - G(x))2^{i-1} \tag{5.4}$$

A number of algorithms based on PCA (Principal component analysis), LDA (Linear discriminant analysis) and ICA (independent component analysis) have been used to address the issue of face recognition. The facial recognition problem is usually effected by the illumination problem in unconstrained environments. In case of low luminous environments, subject faces will have difference of illumination on different points. Compared to other algorithms, LBPH has greater tendency to resolve the illumination problem.

a) Deep Neural Network

Deep neural networks have many algorithms, but for this research YOLO is expected to be appropriate because of its promising results in object detection. The YOLO architecture is more like a R-CNN (fully convolutional neural network) and passes the image (nxn) once through the R-CNN and predict the output (mxm).

Thus, the architecture is splitting the input image in mxm grid and for each grid generation 2 bounding boxes and class probabilities were generated. Note that the bounding box is more likely to be larger than the grid itself.

b) Modification in CNN

The convolutional neural network is used for the detection of an object and achieved results for object detection that are remarkable, but this needs an enormous amount of data[83]. Currently, the object detection was trained on COCO and imagenet data. These datasets have massive data of different objects. The architecture of YOLO shows that it consists of 106 layers for object detection. YOLO (you only look once) [7] is a one stage object detector proposed by Red-

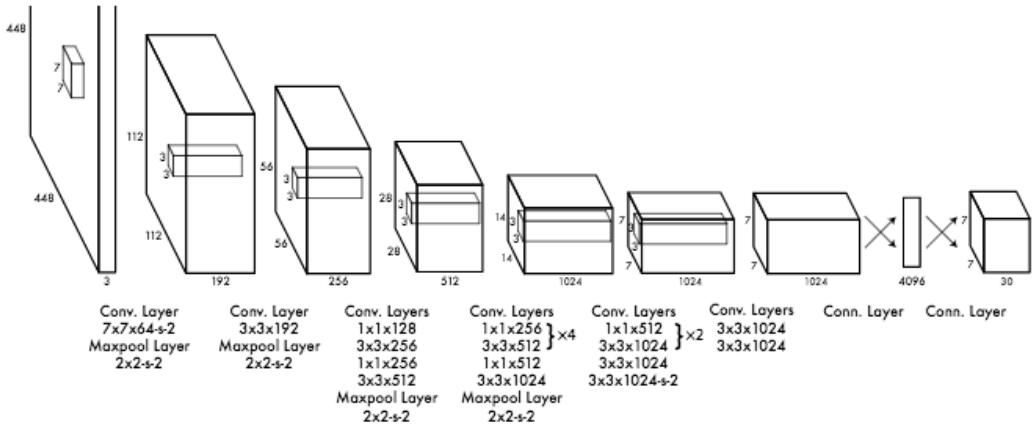


Figure 5.6: YOLO Architecture [7]

mon et al. after Faster R-CNN [84]. The main contribution from YOLO is the detection of full image and webcam. Firstly, it is due to the this pipeline only predicts less than 100 bounding boxed while the Fast R-CNN using selective search predicts 2000 region proposals per image. Secondly, in YOLO frame detection is a regression problem, so a unified architecture can extract features from input images straight to predict bounding boxes and class probability.

YOLO pipeline first divides the input image into an SxS grid, where a grid cells are responsible to detect the object whose centre falls into. The

confidence score is obtained by multiplying two parts, where P(object) denotes the probability of the box containing an object and IOU (intersection over union) shows how accurate the box contains that object. Each grid cell predicts B bounding boxes (x,y,w,h) and confidence scores for them and C-dimension conditional class for C categories. YOLO layer id defined under one by one.

To achieve real-time face recognition with the help of CNN, it is essential to calculate how many numbers of layers are there. Datasets with faces in the wild are limited so to achieve robust face recognition it is crucial to learn parameters from the test dataset.

Following is the description of the layers to design a model for face recognition:-

- Convolutional Layer: The convolutional layer is the vital layer in a CNN. The convolutional layer takes "l" feature maps as the input and has "k" feature maps as output. The filter size is "n*m."

To recognize a face from the image number of features extracted by a layer is essential. To achieve reasonable accuracy it is important not to have an overfitted model. The number of parameters extracted by CNN has a significant role in the overfitting and underfitting of an algorithm. The input features l=32, k =64 as output features combined with filter size n=100 and m=100. The total number of parameters will be calculated from eq (5.5)

$$P = (\mathbf{n}^{*}\mathbf{m}^{*}\mathbf{I} + \mathbf{1})^{*}\,\mathbf{k} \tag{5.5}$$

Where **P** represents the parameters, **l** is feature map of input, **k** is feature map of output and n$\times m$ defines filter size.

- Pooling Layer: The pooling layer is used to reduce the image dimension for further processing.
- Fully Connected Layer: In this layer, all input units have a separate weight to each output unit. To calculate the size of first fully connected layer it is

vital to calculate the size of each convolutional layer. To calculate the size eq (5.6) will be used:

$$\text{Size of layer} = \text{input Size} \times (\text{filter size} - 1) \tag{5.6}$$

Over-fitting or under-fitting are the two important causes of poor performance in machine learning algorithms. These factors will affect the model, which results in poor performance when the model is evaluated on new data. There are three noted reasons for the over-fitting or under-fitting of models.

- Size of data available for training.
- The number of iterations (Training iterations).

There are two approaches used to overcome the fitting problems. One approach is to optimize the number of iterations of training images. To discover the best fit for the number of iterations, the following equation potentially gives us the number of iterations concerning the images in the training set.

From equation (5.7) calculate the number of iteration required with respect to the number of training samples.

$$\begin{aligned} f(x) = \frac{129397171x^4}{472955617158528} - \frac{101863288699x^3}{236477808699x^3} \\ + \frac{640697052227x^2}{3477614832048} - \frac{213868739879995x}{7389931518102} \\ + \frac{56865390110280}{28643145419} \end{aligned} \tag{5.7}$$

The second approach is to randomly drop nodes from the network to reduce the parameters. During training of Neural network, certain numbers of output layers was ignored. The dropout model will be used to avoid over-fitting and under-fitting problem.

$$E_D = \frac{1}{2}\left(t - \sum_{i=1}^{n} \delta_i w_i I_i\right)^2 \tag{5.8}$$

Equation 5.8 Dropout rate is δ where $\delta \sim$ Bernoulli(p). This means a is equal to 1 with probability p and 0 otherwise.

5.2.3 Databases

In face-recognition and face-recognition research, it is common practice for researchers to use a collection of generic and widely accepted facial databases for their research. This is done so that the work and results are acceptable in the research community. By choosing the standardized database it is very easy for the researchers to replicate and verify any previous results. For the described research it is important to have two types of data sets, one to train the model and the other(video) to test the accuracy.

To evaluate the accuracy and time complexity; it is important to use a standardized dataset. YaleB is the most significant database with frontal faces while considering different lighting conditions and pose variations. There are single light source 5760 images of 10 subjects each seen under 576 visual conditions included in the YaleB database (9 poses x 64 lighting conditions). A picture with ambient (background) lighting was also recorded for any subject in a specific pose. Therefore, the cumulative number of images is currently 5760+90=58505=5850 [79]. Number of samples are presented in figure 5.7.

5.2.4 Accuracy Metrics in Machine Learning

Accuracy of facial detection and identification in poor illumination can be achieved using a number of techniques. One of the important methods is to build a confusion matrix [85]. The confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the actual values are known. It allows for the visualization of the performance of an algorithm. The confusion matrix gives an insight of error concerning the class for the TP, FP, FN, and TN which are True positive, false positive, False-negative, and True negative. The confusion matrix contributes to finding the Accuracy, Recall,



Figure 5.7: Face images from the Extended Yale Face Database B database under different illuminations.

Precision, and F-measure. Figure 5.8 defines the confusion matrix.

- Accuracy or classification rate is defined by:

$$Accuracy = \frac{TP + TN}{TF} \tag{5.9}$$

- The precision is determined by dividing the total number of True Positive Examples classified by the total number of samples.

$$Precision = \frac{TP}{TP + FP} \tag{5.10}$$

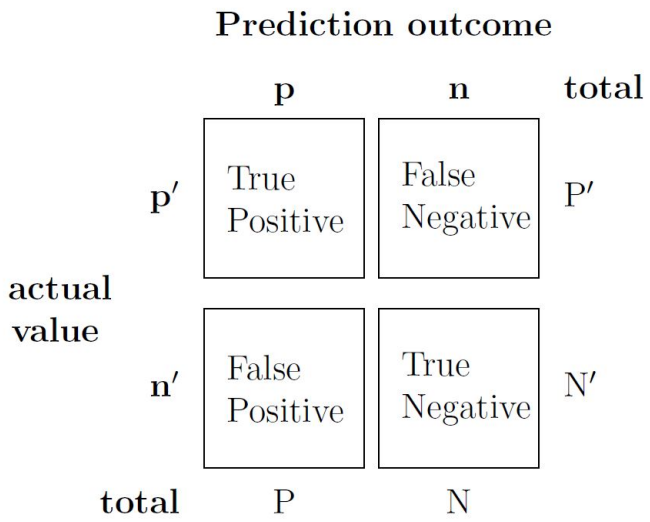


Figure 5.8: Confusion matrix

- F-Measure based on Precision and recall. The F measure will always be nearer to the smaller value of Precision and Recall.

$$F - Measure = 2 \times \frac{Recall \times Precision}{Recall + Precision} \tag{5.11}$$

  Receiver Operating Characteristics (ROC) is a graph used to visualize the performance of the system [86]. It is a distinct option for precision–recall curves. It is a graphical representation for displaying the transition between TP and FP.

The comparison of LBPH with KNN and YOLO with RGB and Grey-scale images will be made to observe the accuracy and speed of recognition in a low light environment. For Accuracy, Precision, recall, F-measure, and ROC results, Values of TP,TN,FN, and FP will be calculated using a Confusion matrix.

5.3 Conclusion

Chapter 5 explained the research methodology adapted for this research. A technical introduction to the methodology, along with the modifications proposed for YOLO training was also stated. The illumination conditions of the dataset and confusion matrix used to evaluate the results was also discussed in this chapter.

Chapter 6

Implementation of Machine Learning Algorithms for Face Detection and Recognition

6.1 Introduction

The research was aimed to achieve higher face recognition in low illumination environments. For the precision of the face recognition system, selection of a dataset composed of low illumination images is very critical. The Yale Face database B was then chosen. The expanded Yale Face Database B comprises of photographs with varying settings for lighting. The images with different illumination were used as training and testing sets of the machine learning algorithms. Firstly, as a benchmark, the classical approach of LBPH with KNN was implemented to compare with the proposed YOLO.

To determine the accuracy of different techniques, the recognition rate (RR) was calculated using the equation 6.1.

$$RR(\%) = \frac{\text{Number of correct match}}{\text{Number of total number of images}} * 100 \tag{6.1}$$

6.2 Extended Yale Dataset B

Algorithms for machine learning are highly dependent on the quantity of data. Data quantity is now one of the bottlenecks among a host of concerns in machine learning algorithms. Machine learning methods, such as deep neural networks, need a vast amount of data to automatically create features. There are many techniques based on deep neural networks like Fast R-CNN and YOLO. Face recognition with deep neural networks also requires a large amount of subject data. Image augmentation is used to create a large dataset for YOLO.

The images in the database were collected using a purpose-built illumination rig. The rig has 64 computer-controlled strobe modules. Different poses of the subject were taken with the help of this illumination rig. For this research, subject images are divided into five subsets with increasing illumination. Figure 5.7 show the few subjects from the dataset [87].

One of the most challenging parameters in face recognition is the variation in illumination on the subject because it complicates the extraction of the features from the images. Extended Yale face Dataset B is a best-fitted dataset for this purpose. In the extended YaleB dataset, the camera axis has been changed each time to increase the angel of light incidence on the object to present more challenging light conditions. The naming convention used in the dataset represents the angle of falling light on the image. The positive implies that the light source was at the right side of the subject while the negative means it is below the horizon. The angle of incidence on the image is very important because the lighting conditions will define the angle of the light source and the image. Image 1,2,3,4 and 5 were about 12 degrees front from the optical axis, while 6, 7, and 8 were about 24 degrees. The axis and angle of lighting in the image are defined in figure 6.1 [88] [89].

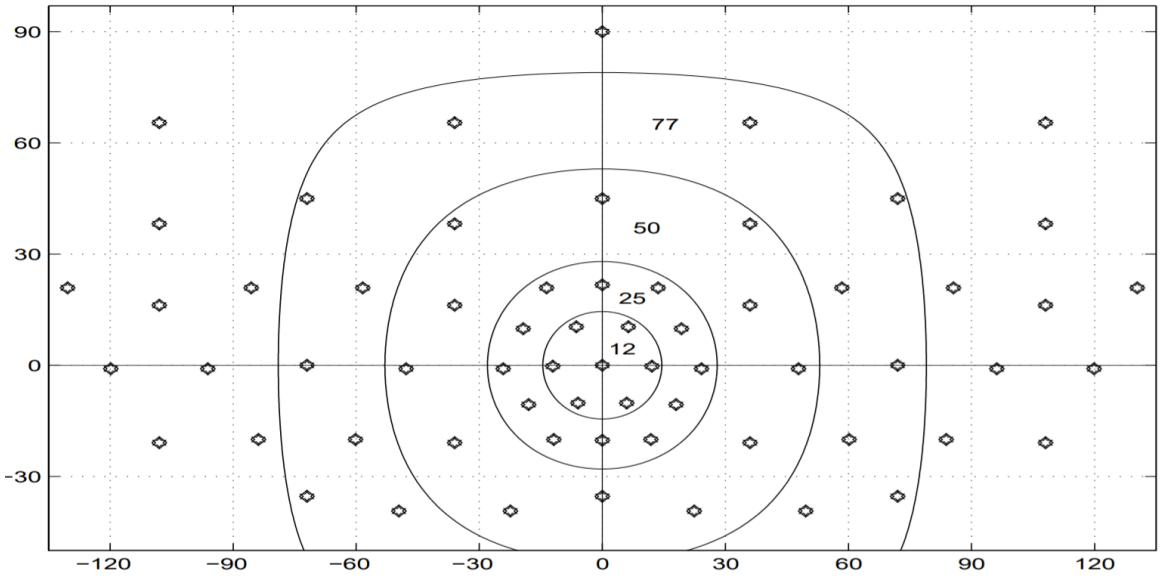


Figure 6.1: Configuration of nine light source directions with all 200 sample in Extended Yale dataset B [10]



Figure 6.2: Different angle of lighting on subjects of YaleB dataset

6.2.1 Classification of Extended YaleB Dataset

To validate the algorithm results on different illumination, Extended YaleB dataset was divided into five different subsets with respect to different illumination level. It is a common practice to classify the images with lower angles as illuminated and images with higher angles as ill illuminated. To determine the characteristics of images, statistical image visualization was used. Statistical visualization of images is an efficient way to characterize and explain many statistical details of images required for their analysis. The histogram is one such method and can be used to extract many details about images such as exposure, contrast, dynamic range and saturation.

A histogram is a statistical representation of an image. It does not show any information about the location of the pixels in the images but a histogram represents the number of pixels for each intensity value in an image. Figures 6.3 show the histograms for images of subset 1 to 5 of subject 1.

The image histogram represents the distribution of different intensity pixel counts present in an image. From the pixel count brightness of the image was generalized, the distribution of pixel intensity values from left to right represented the distribution of illumination distribution. This helped to divide the image into illumination dependent subsets. The subsets of the frontal face extended YaleB dataset which was divided according to the difference in angle and lighting, as seen in figure 6.4:

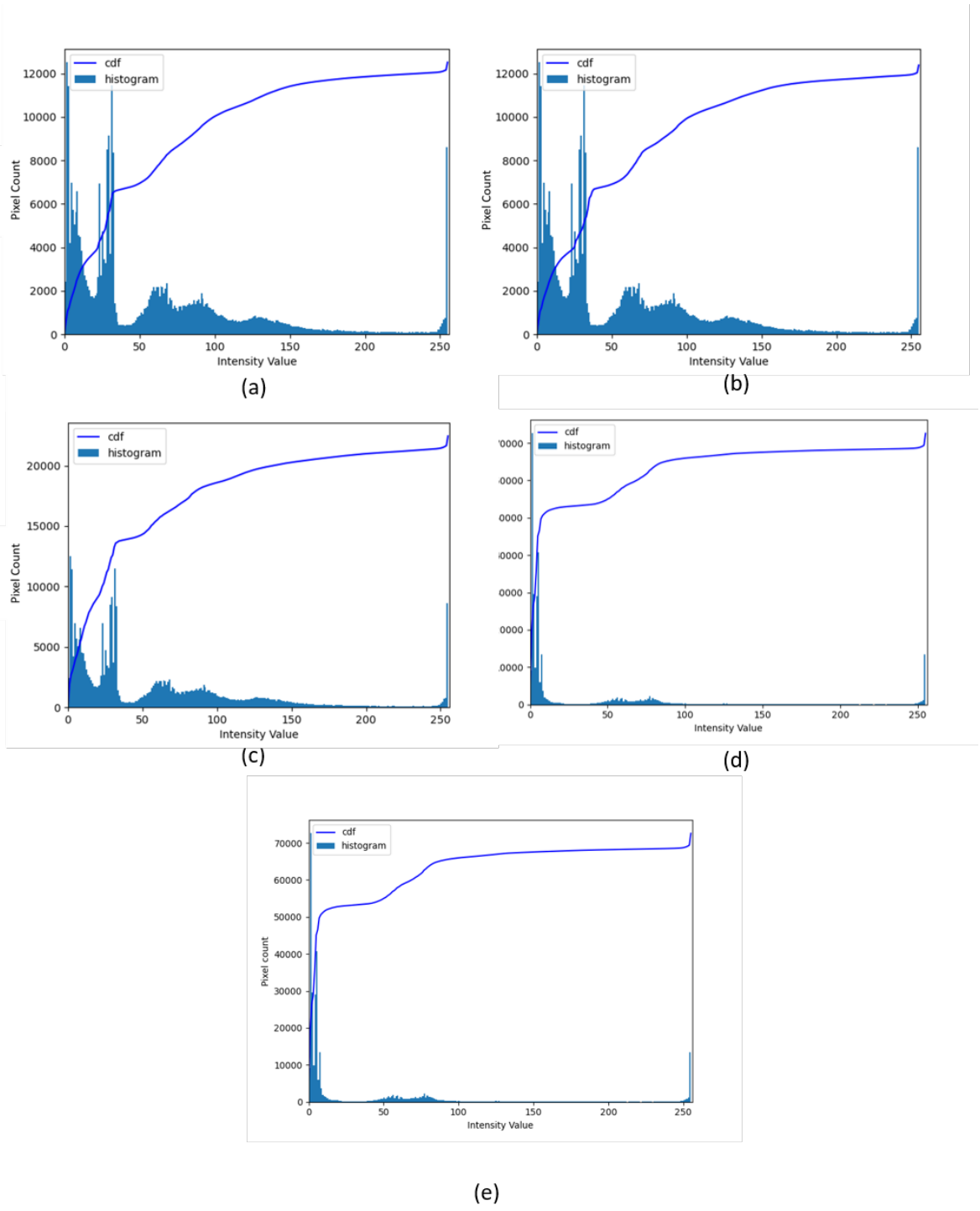


Figure 6.3: Histogram and CDF for (a)Subset 1, (b)Subset 2, (c)Subset 3, (d)Subset 4, (e)Subset 5

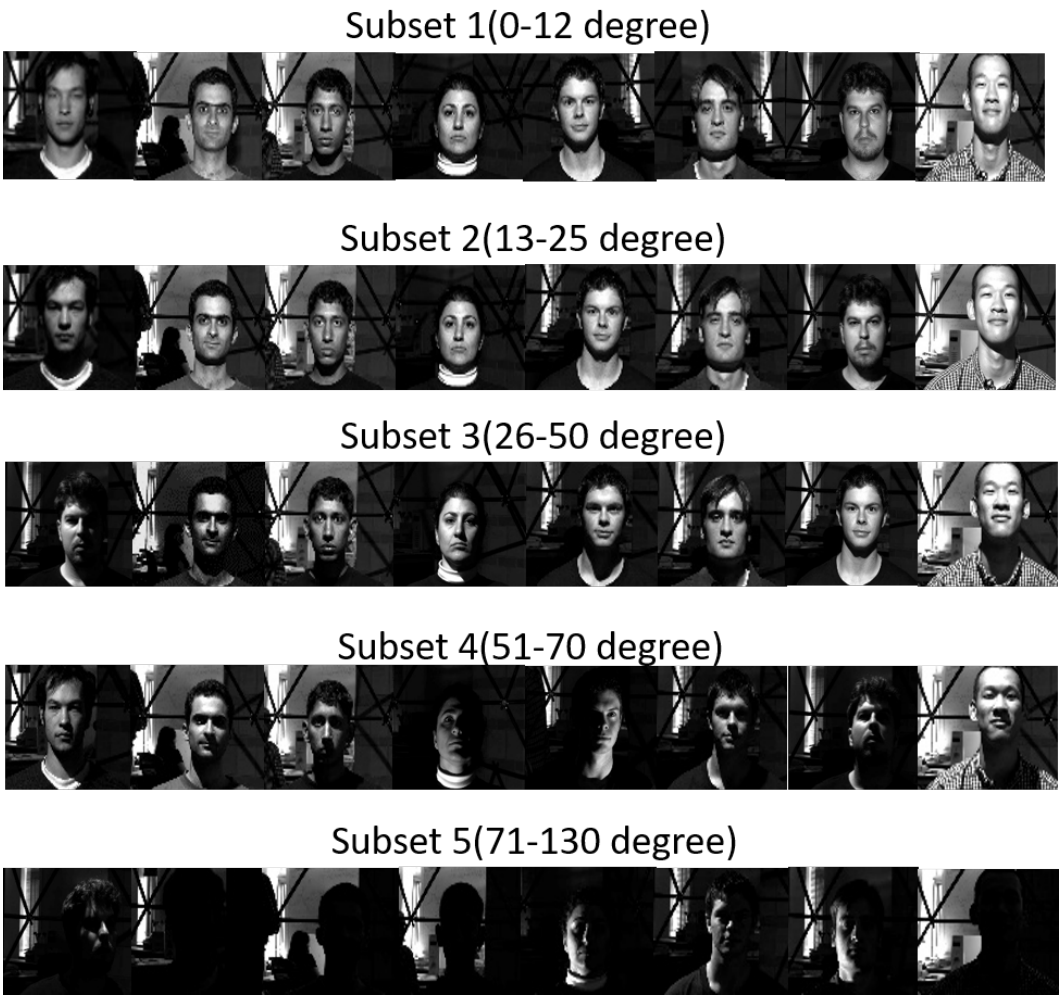


Figure 6.4: Different illumination subsets in the Extended YaleB face database.

These five subsets of the dataset were used as training and testing of the algorithm. For LBPH and KNN, 80 percent of these images from every subject was used as training of the algorithm. However, the vast training set was needed for the proposed approach. For training a neural network, a minimum of 1k images for each subject were required. To increase the images in the training dataset, it is essential to implement data augmentation.

6.2.2 Data Augmentation

Data augmentation is a synthetic technique of creating a repository of data. The process increases the diversity of the data available for training the model. Data augmentation also enhances the performance of a deep learning model by increasing the number of samples at each stage of training and this helps reducing the variance of the model by making it better at generalizing. The Keras library was used for data augmentation in this research [90]. While doing data augmentation, important parameters are delta , shear range, zoom range and rotation. For following data augmentation image generator has following properties.

rotation range = 40

width shift range = 0.2

height shift range = 0.2

shear range = 0.2

horizontal flip = 0.2

fill mode = Nearest

Data augmentation helps in increasing the overall diversity of the dataset by increasing in different number of samples of different angels. Figure 6.5 shows the result of data augmentation for subject 1. For every subject in training data augmentation was used in same manner.



Figure 6.5: Data Augmentation for subject 1

6.3 Implementation of LBPH with KNN

In this research, LBPH with KNN approach was implemented as a baseline approach for face detection and recognition. LBPH was used for the extraction of features and KNN was used as a classifier. By splitting the image into smaller regions, the purpose of LBPH is to reduce the proportions. To find the distance between the nearest neighbours, Euclidean distance was used. Figure 6.6 describes the flow graph of the LBPH with KNN approach.

Figure 6.6 illustrates the steps of face detection and recognition with a classical approach. This method of recognition consists of three modules.

- Preprocessing of dataset
- Face image training module
- Face recognition Module

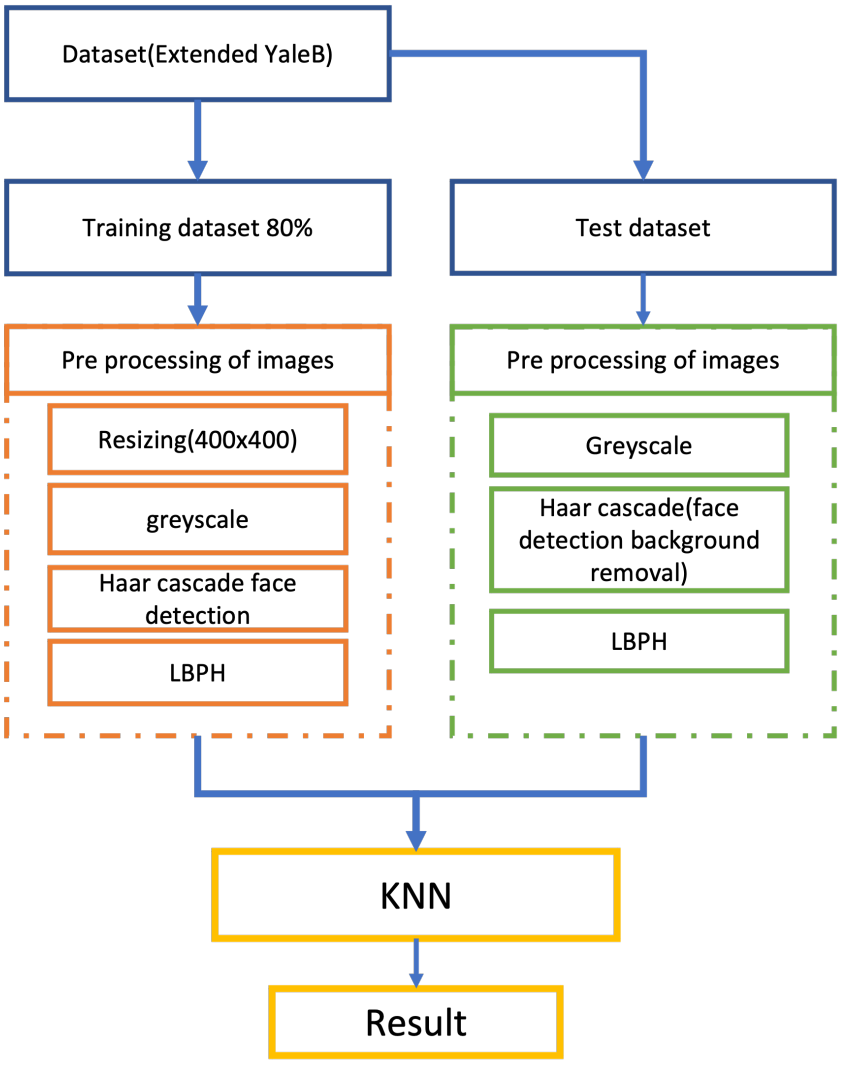


Figure 6.6: LBPH and KNN implementation

Preprocessing of the image in LBPH with KNN and YOLO techniques are very important in reducing the computational cost of the algorithm. The preprocessing of images requires the resizing of images to erase the background from the images, accompanied by cropping the face. By using Haar Cascade face detection, cropping was achieved. Haar Cascade is a method of target identification used by Viola and Jones. It is a methodology based on machine learning, where a cascade function is equipped with the help of positive and negative images. It is then used in other images to detect objects. To create a stable recognition scheme, it is necessary to remove the context and transform the image to a grey-scale. Figure 6.7 shows the resultant image of one subject before and after preprocessing.

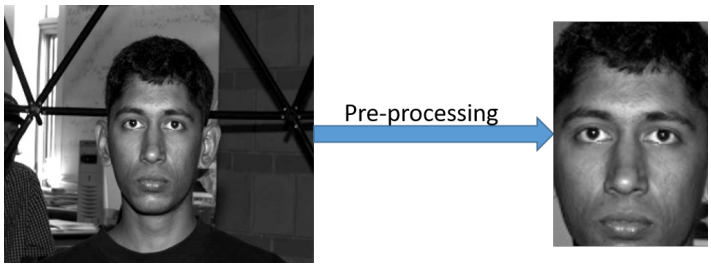


Figure 6.7: Pre-processing of image using Haar-Cascade Classifier

As in the first module, background removal was done using the Haar cascade face detector, which helps in cropping the face from the image. After applying Haarcascade, the new face representation obtained from LBPH represents the face in a smaller dimension. Finally, calculated the similarity between the source image and the testing image with KNN.

6.4 Implementation of YOLO

When a human being looks at an object, the brain responds to the main features of the object quickly and accurately due to the properties of the neural structure present in the brain. Similar to the working of the brain, various algorithms use the neural network structure to achieve the goal of object detection and recognition. These algorithms are grouped as convolutional neural networks in the field of machine learning [91]. To detect the object in an image, a classifier is used

which takes the particular object and evaluates it at various location in the input image. Models like deformable part models (DPM) and YOLO uses the approach of a sliding window in which the classifier runs on the entire image to locate and classify the image.

Region-Based Convolutional Neural Networks(R-CNN) uses a convolutional neural network architecture that processes a region of interest from the input image, reducing the processing time. This method first generates the bounding boxes in an image and then runs the classifier on the proposed boxes. After the classification, post-processing is done on the input to remove the identical bounding boxes and re-score the boxes concerning the object present in the image. The computational complexity for CNN is higher than R-CNN, but results in terms of accuracy are better than R-CNN. To improve the accuracy and reduce the processing time complexity of R-CNN, You only look once (YOLO) is used. YOLO is a deep learning technique in which the training for object detection is carried out using a loss function. YOLO uses re-framing of the object as a single regression problem in which object detection is achieved using image pixel to bounding boxes and finding class probabilities using region encapsulated in the bounding box. Figure 6.8 explains the architecture of YOLO for face detection and recognition.

YOLO inspects the entire image at once during testing and it processes 45 fps, unlike a sliding window technique. Training in YOLO feeds on annotations extracted from the bounding boxes. After this, YOLO would be able to encode the contextual information about the appearance in terms of classes[92]. Fast R-CNN gives background errors more than YOLO because, unlike YOLO, Fast R-CNN is unable to see the broader context in the image [93]. After all, it does not follow the technique which scans the entire image at once.

The YOLO design does the training end to end and achieves the real-time speed. The most important part of developing and analyzing such a system is a dataset. CNN needs massive data to train the network. Several datasets are consisting of images which help to evaluate the object detection. For face detection and recognition in a low illumination environment, it is essential to use face datasets with different lighting conditions. YaleB extended was used for this

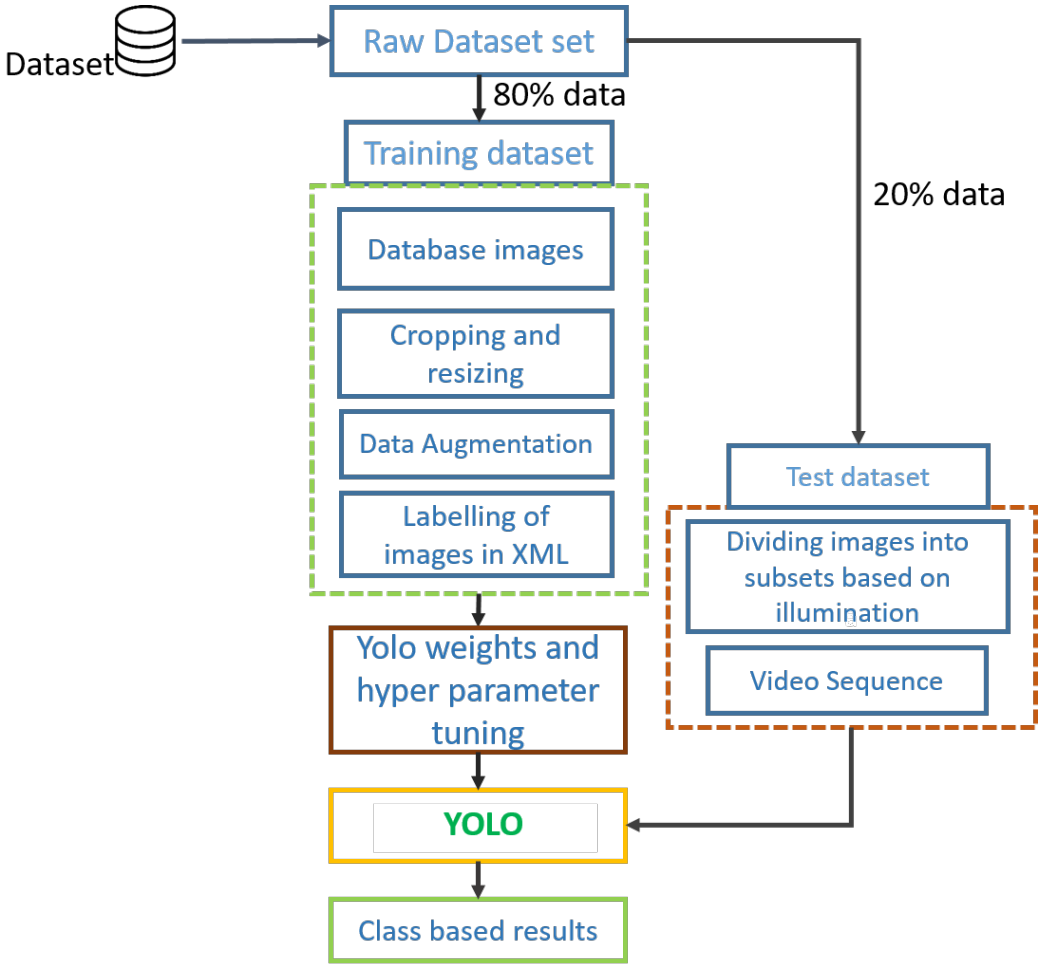


Figure 6.8: Framework of Yolo implementation

research. YOLO network feeds on the annotations for training. To annotate the images bounding boxes were drawn on the images. These bounding boxes are the ground truth of the face images, and the coordinates were saved into .xml format. Figure 6.9 shows the bounding boxes in the dataset. .xml for creating the bounding box is presented in chapter 8.1 Appendix.

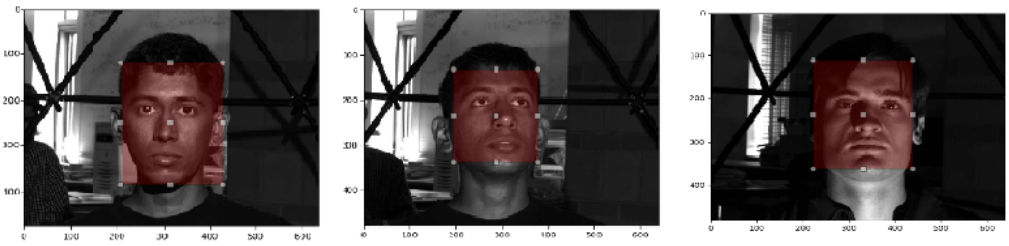


Figure 6.9: Bounding boxes on subject images

6.4.1 Data Augmentation for YOLO Implementation

Training, validation and testing is separated into three subsets. 80 percent of the data collection consisted of a research dataset, and 20 percent of the dataset consisted of a review dataset. Neural networks need a vast dataset to train themselves but the distribution of the dataset into three parts reduces the size of the training set. To overcome this problem of small dataset, several techniques such as data generalization, regularization, and data augmentation are available. The data augmentation technique is generally a preferred approach in which new data points are created by manipulating the original data [94]. The manipulation can be done by rotating, resizing, cropping, random pixel erasing, etc. The process increases the diversity of the data available for training the model.

Data augmentation also increases the performance of a deep learning model. In this research, Keras library was used for data augmentation using python. After performing data augmentation on the dataset, the images were labeled with the subject name and saved in XML format. Images from the dataset and .xml annotated file were used for the training of the network. .xml defines the dimension of the object place in the image. The annotated file name must be same as the image present in the training dataset folder. The darknet framework was used to train YOLO which is open source platform written in C. Training of the network depends on the number of training samples and iteration over the testing samples. The selection of the number of iteration and the number of testing samples were selected empirically to optimize the network.

6.4.2 Hyperparameters for YOLO Implementation

The deep neural network is full of hyper-parameters [95]. Finding the optimal values of these hyperparameters in a high dimensional space is not a trivial challenge. The learning rate, batch size, and scale-ratio of anchor boxes are hyper-parameters. These parameters act like a knob in training the R-CNN hence finding optimal values to enhances the performance of R-CNN is very important [96].

There are various ways to calculate the sizes of bounding boxes for detection tasks. One approach is to empirically predict the bounding box values. But this approach is prone to errors. An alternative approach is using template bounding box dimensions. K mean clustering will be used to detect a good candidate for the anchor box.

The most important hyperparameter in face detection is the "scale and ratio of anchor boxes". The shape of anchor boxes affects the overall performance of the algorithm. The performance of YOLO decreases as it detects small-scale objects in a scene. To avoid this problem, anchor boxes are modified. For general detection, like in an Alexnet network, it contains all sizes of anchor boxes for possible items. A large number of anchor boxes increases the complexity and it also do not detect the small objects or the background scenes (very big objects). A large number of anchor boxes are helpful in some cases like detecting the sign boards or detecting cars in a scene but the error rate is higher in case of face detection and recognition.

The aim to select anchor boxes is to select a valid ROI (region of interest). Intuitively, for most faces in the dataset the height of the face is always larger than the width. Therefore the selection of anchor box parameter for face detection should not be the same as for general object detection [97].

The approach uses template bounding box dimensions. To find the best template for anchor boxes, exploratory data analysis and K mean clustering is used to detect good candidates for the anchor box. Exploratory data analysis (EDA) helped to analyse the data in a good way. In general, bounding boxes

for objects are given by tuple in the form x_0, y_0, x_1, y_1 where x_0 is x-min, y_0 is y-min, x_1 is x-max; y_1 is y-max.

Yolo object detection was done by using class score. Same technique will be implemented for face recognition with Yolo. Every subject from the database was given a class name. The end result for Yolo will be calculated score on the basis of their input images.

6.4.3 Exploratory Data Analysis

Exploratory data analysis (EDA) is perhaps the most important step in the machine learning algorithm. EDA was done on data annotation files. Those files are converted into .csv file to gather all data points to calculate a good candidate for anchor box. To do EDA, x-min, y-min, x-max, and y-max points of training dataset were used. EDA visualization explains the existence of data point in the dataset. Visual representation of data is shown in Figure 6.10 and 6.11.

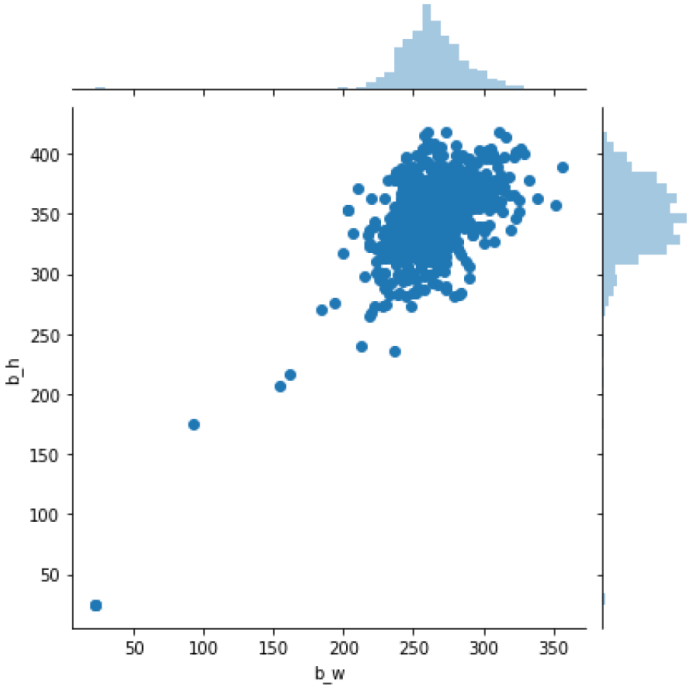


Figure 6.10: Plotting data points of training dataset

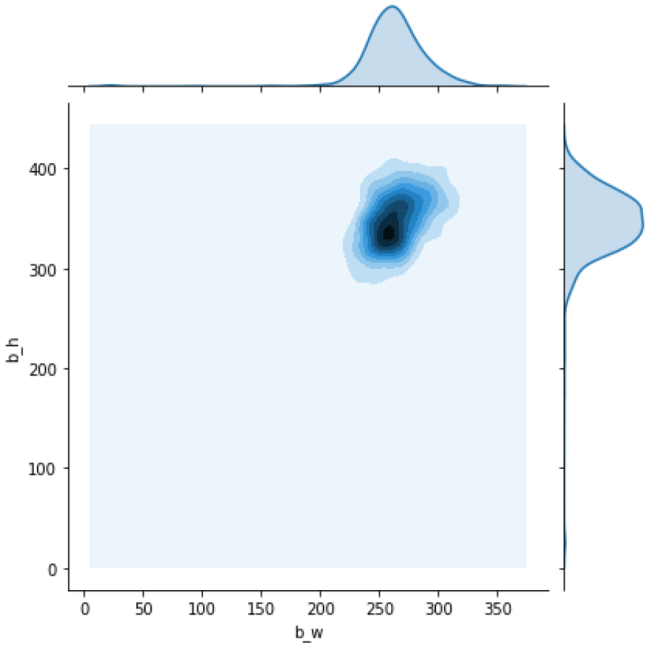


Figure 6.11: Kernel density estimation of data points

where b_w is width and b_h describes the height of the bounding boxes drawn from dataset.

After visualizing the dataset, K mean clustering was used to find the mean of every cluster. This number of anchors was dependent on the number of clusters. From the data, it is evident that most of the bounding boxes are large in size. The value of K selected as 5. Figure 6.12 shows the K mean clustering for the training set used for this research.

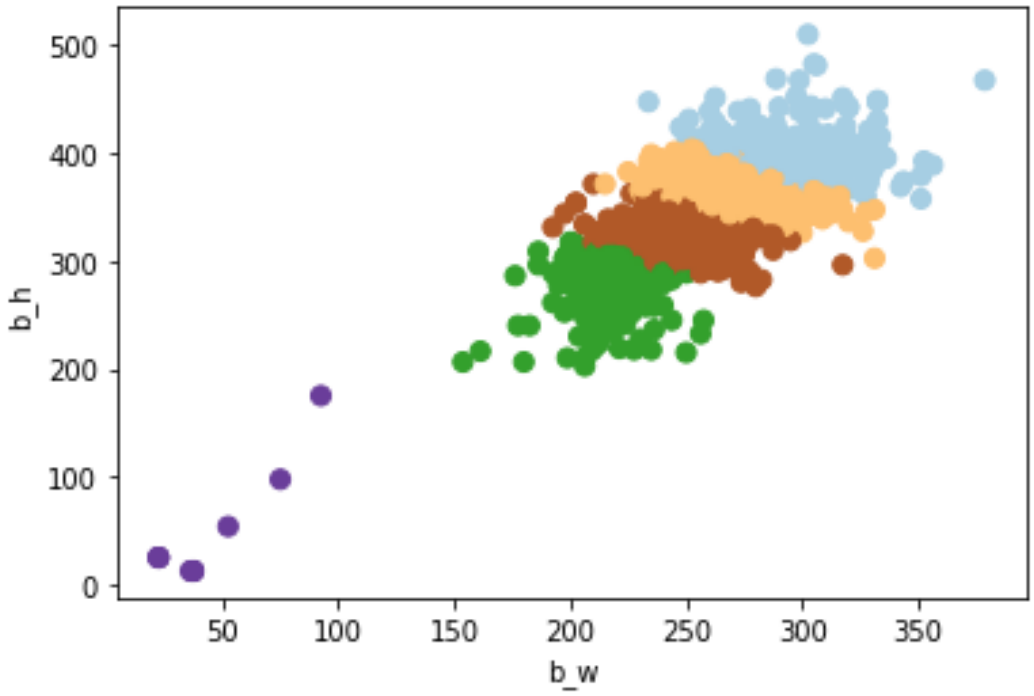


Figure 6.12: K mean clustering using k=5

For clustering, the mean size of boxes from each cluster were extracted. It was observed that the shape of the boxes are slim. Slim refers to the fact that height of the bounding box is also greater than the width of the box. The original YOLO had flat boxes, which means the height of the box was less than the width of the box. These boxes create the localization error in case of faces for the used dataset. The mean values of height and width of the bounding box of faces are:[262.5 376.25][246.25 332.5][305. 410.625][292.5 372.5][221.25 288.75][276.25 343.75].

6.4.4 Selection of Anchor Boxes

The bounding boxes used in training are slim and big in shape. Keeping in view all parameters of bounding box, the shape of the anchor boxes are (30, 61); (45, 62); (59, 119); (90, 116); (156, 198); and (326, 373). The scale and shape of anchor boxes increase the face detection in the frame. Along with selection of

anchor boxes, the batch size controls the accuracy of the estimate of the error gradient when training neural networks [98]. Other selected hyperparameters for this experimentation are shown in table:

Table 6.1: Hyperparameter for YOLO

Hyperparamters %	Values
Batch size	64
No of Classes	27
Anchors size	(30, 61); (45, 62); (59, 119); (90, 116); (156, 198); and (326, 373)
learning rate	0.001

The hyperparameters in neural networks plays an important role in training. The most important among all are batch size and learning rate and there is a trade-off between these parameters. Batch size is a term used in machine learning and refers to the number of training examples in one iteration.The batch size is also dependent on the computational power. The other parameter which also plays an important role in training is the learning rate. Usually, the large learning rate can cause the model to converge too quickly to a sub-optimal solution. A traditional default value for the learning rate is 0.1 or 0.01, and this may represent a good starting point on your problem [99]. The small batch size and lower learning rate will leads to overall higher accuracy i-e neural network that perform better in the same amount of training dataset [100].

6.5 Conclusion

Chapter 6 describes the following steps in implementation of the classical approach which is LBPH with KNN and proposed YOLO-R-CNN based approach. These steps include Data preprocessing, Data Augmentation for the proposed method, and annotation of images for YOLO training purpose. Preprocessing of data describes how the data was divided into different subsets. After this data processing,

the selection of hyperparameter for YOLO was discussed. The most important hyperparameters are the selection of anchor boxes, batch size, and learning rate of the model. Anchor size is responsible for detecting subjects in the frame and batch size was used in training. In next chapter, Results from these implementations will be discussed.

Chapter 7

Results and Discussion

7.1 Introduction

The results of the implementation of algorithms using a low illumination dataset are discussed in this chapter, followed by the statistical evaluation of the performance of the algorithms. Firstly, LBPH with KNN using the ORL dataset is discussed and afterwards LBPH with KNN and YOLO results is presented. Lastly, the results of face detection and recognition using preprocessing of low illumination images are given.

7.2 LBPH with KNN using ORL Dataset

These research works explain results using different databases, like Feret, ORL, LFW, and YaleB [101]. Every database has images with different specifications like the angle of lighting and image sizes. Implementation of LBPH and KNN on the ORL dataset has shown promising results. ORL dataset consists of frontal face images and has good lighting conditions with (0° pan). The recognition rates, as defined in equation 6.1, for LBPH and KNN for the ORL data set using different training set percentages per subject are shown in table 7.1:

There are different distance matrices such as the Euclidean, Correlation,

Table 7.1: The recognition rate using LBPH and KNN on ORL dataset

LBPH+KNN(ORL dataset)	
Training & test dataset %	Recognition rate(%)
50% 50%	94%
70% 30%	97%
90% 10%	99%

Canberra, and Manhattan distance, which can be used to find the closest neighbor in KNN [102]. The results of LBPH and KNN, based on these matrices, are given in figure 7.1:

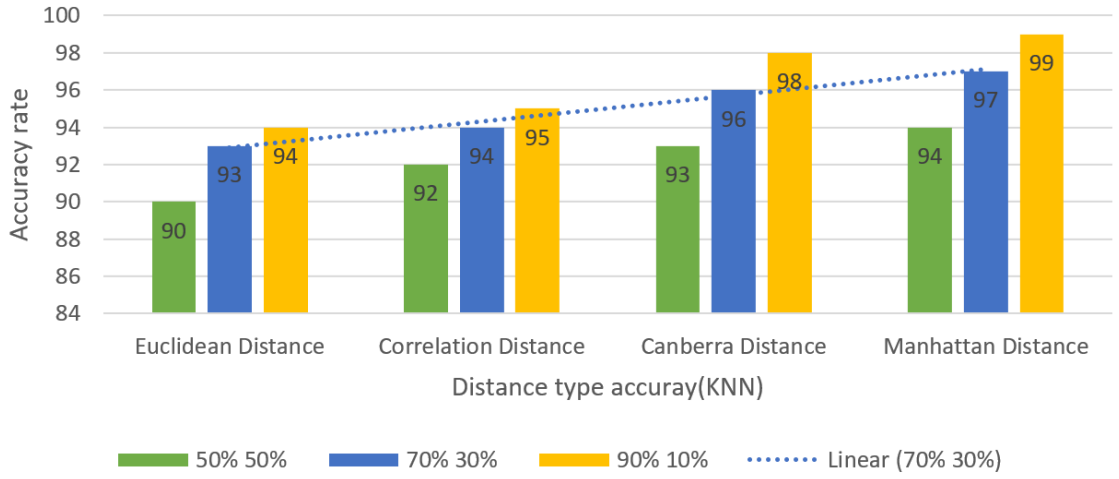


Figure 7.1: Accuracy with different distance method in KNN

Different distance measures for KNN result in different accuracy on the ORL dataset. These KNN distance measures were implemented on the frontal face ORL dataset with no variation in lighting. In all distance measures, Manhattan distance shows higher accuracy, which is 99%. To analyze the results with respect to different lighting conditions, it is essential to have a dataset with different light variations and poses.

7.3 LBPH with KNN using the Extended YaleB Dataset

To do the experimentation with LBPH and KNN, it is essential to prepare the data for training purposes. Preprocessing includes the background subtraction of the image. To achieve background subtraction, a Haar Cascade detector was used, which detects the face from the image and crops the face. These cropped images were used as a training set. To determine the accuracy of the algorithm, test images from the remaining dataset were used. It was also observed that the accuracy of LBPH and KNN depends on the number of training samples. In figure 7.2 the result for LBPH and KNN classifier are shown for different training set sizes, and different values of the threshold K. The LBPH with KNN algorithm was evaluated considering the following parameters:

- The threshold K for the distance.
- The number of training samples used per class.
- Varying lighting condition.

7.3.1 Results for Subset 1

Subset 1 consisted of 77 images per subject with an incident light angle varying between 0° to 12°. These images were considered as well-illuminated images.

Figure 7.2 shows the impact of threshold K on the accuracy of face recognition. The parameter k in KNN corresponds to the number of closest neighbors to be used in much of the selection process. The subset 1 used was from the extended YaleB dataset and has proper illumination. As can be observed from the figure 7.2 accuracy increases with the increase in threshold up till the K=2, but increasing the k in KNN also increases the rate of the wrong classification of images. Increasing the number of samples in the training set also increases the

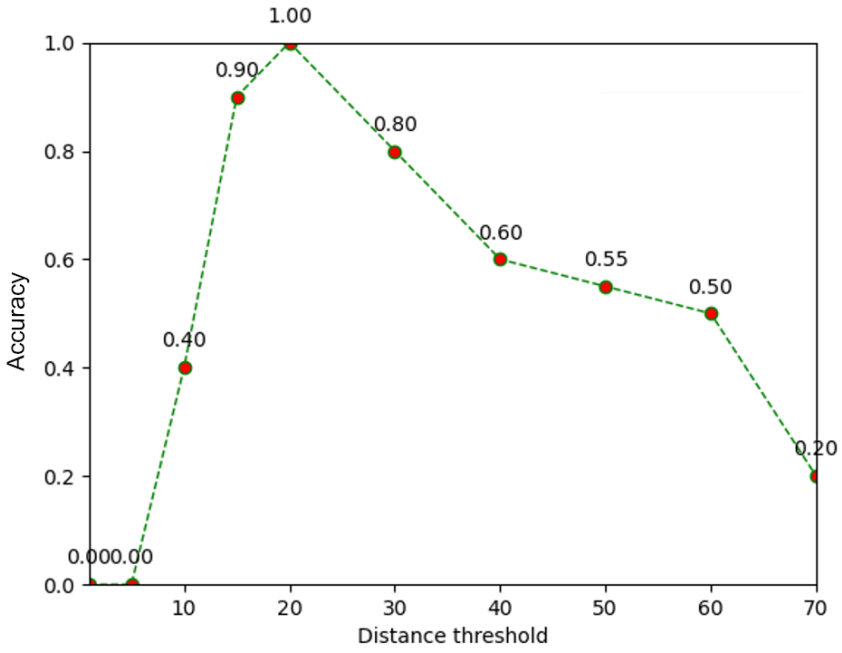


Figure 7.2: Recognition accuracy of LBPH with KNN with different K values

accuracy of recognition. The equation to calculate the accuracy is:

$$RR(\%) = \frac{\text{Number of correct match}}{\text{Number of total number of images}} * 100 \tag{7.1}$$

The variation in precision with respect to the number of samples was also studied, as seen in the figure. 7.2 For example, if the training set consists of 10 sample images and the threshold value is 15, the classifier was able to label 30% images with an accuracy of more than 0.9. The same threshold was applied to 50 images which showed a 0.9 accuracy rate for more than 70% images of the test dataset. Increasing the number of samples to 70 increases accuracy for 99% images.

7.3.2 Results for Subset 2

Subset 2 has 120 images per subject. This subset consists of images that have a light incidence angle from 13° to 25° which were also considered as well-illuminated

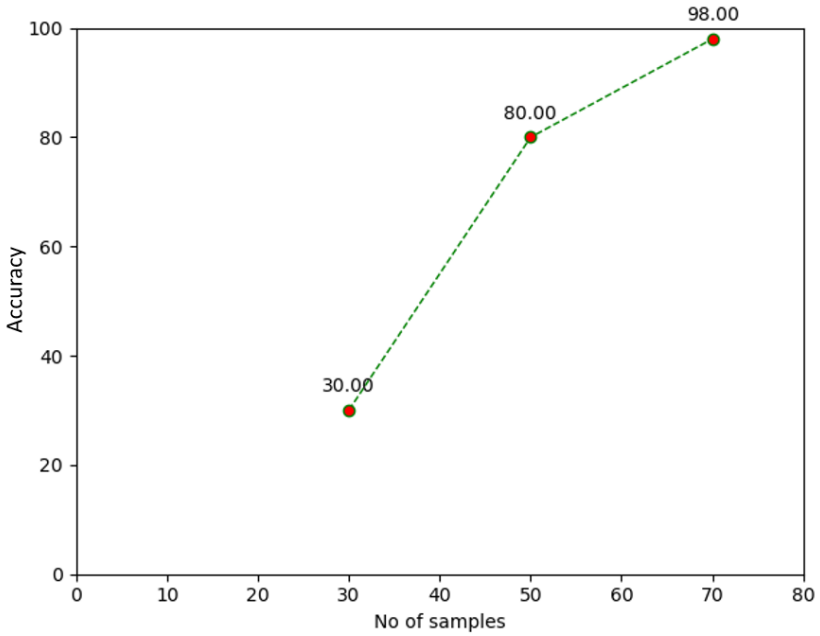


Figure 7.3: Recognition accuracy of LBPH with KNN where angle of light is 0 to 12 degrees(which corresponds to a well illuminated image.)

images.

There was no difference in the results for subset one and subset two. From the histogram representation of subset 1 and subset 2 in figure 6.3, it can be seen that pixel intensity distribution count of both subsets was similar. This is why the accuracy result for both the image datasets are the same.

7.3.3 Results for Subset 3

Subset 3 consists of faces with a lighting angle between 26° to 50°. Previous researches indicate this lighting condition is acceptable. Good lighting condition images are those with less shadow, and the edges are well detected. 10, 50, and 70 samples for the training set were used for this subset. Figure 7.5 shown the results obtained for subset 3.

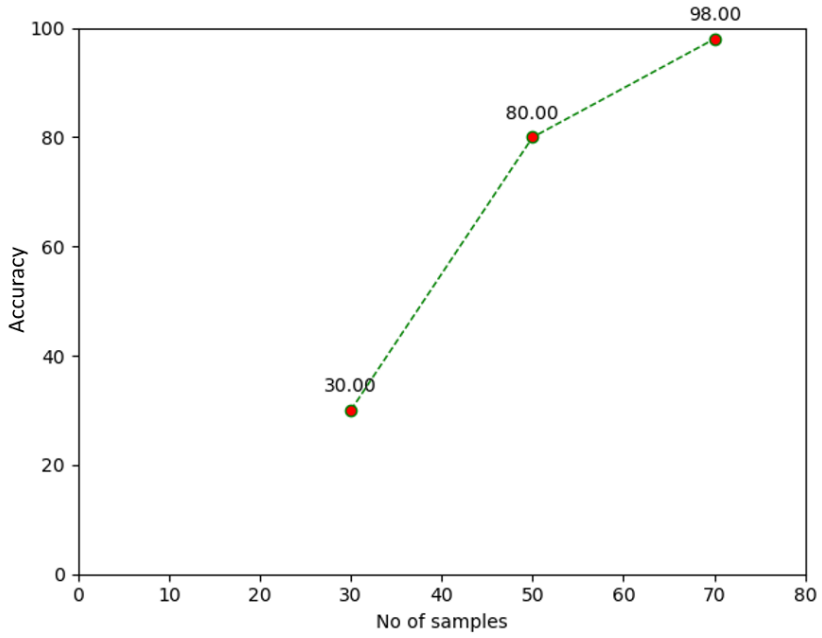


Figure 7.4: Recognition accuracy result of subset 2 using LBPH and KNN

Image with 13° to 25° where the lighting angle is described as well illuminated image

Results express the relationship between lighting and accuracy. The angle of lighting is 26° to 50°. The detection of faces was more than 96%, but it shows classification error for images, which reduces the accuracy.

7.3.4 Results for Subset 4

The subset 4 consists of low light images. The lighting angles range from 51° to 70°. Such images are considered as low illuminated images, and it is also difficult to detect the object (face) in such image. Figure 7.6 explain the result for subset 4 :

The accuracy was low as the lighting condition on the images changes. The accuracy rate falls to 80% for subset 4.

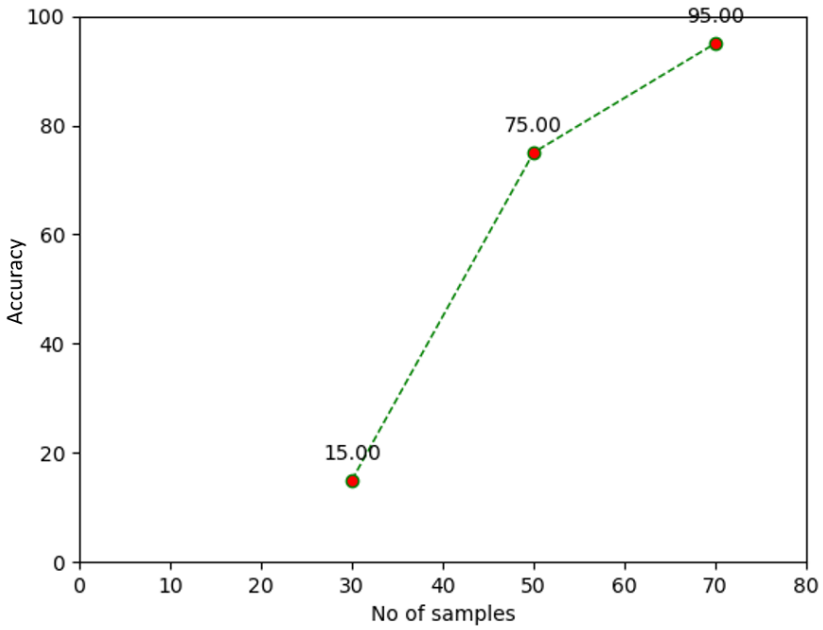


Figure 7.5: Subset 3 Recognition accuracy result for LBPH with KNN

7.3.5 Results for Subset 5

The last subset of the extended YaleB dataset which was used in the simulation is the subset 5, which has the worst lighting due to the large angle of lighting. Results from the subset are shown in the figure 7.8.

The accuracy of face recognition for subset 5 is very low. In subset five, due to the lighting conditions, the error rate was high. Figure 7.8 shows how face recognition for subset 5 depends on the samples of images per subject.

The implementation of LBPH and KNN with the extended YaleB dataset shows that the results are dependent on the lighting condition. As the lighting conditions changes, it affects the accuracy of face detection and recognition. The error rate for low illumination image was higher than the images with good light. The division of images into different subsets with respect to lighting angle (illumination difference) allow this experimentation to be more concise and elaborates

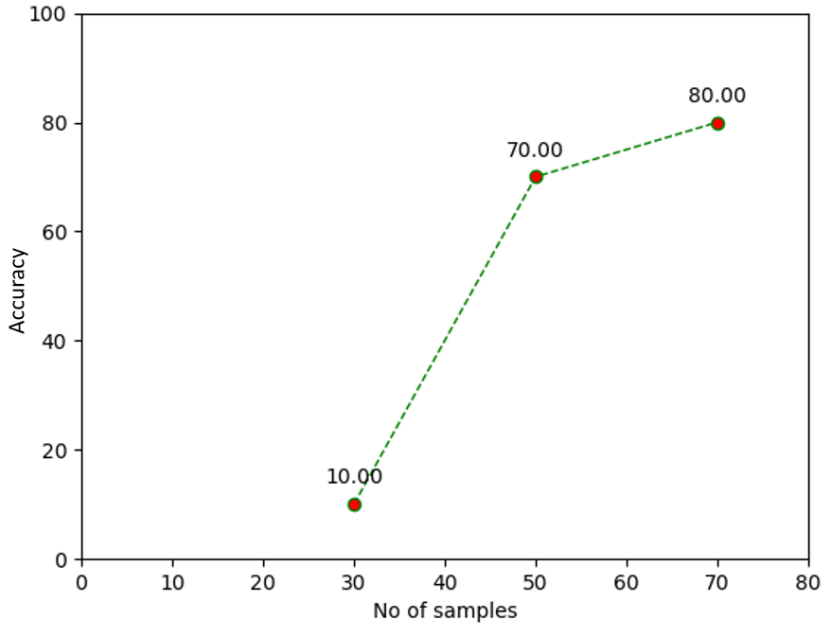


Figure 7.6: Accuracy results for subset 4 using LBPH with KNN

on the low lighting condition on face recognition procedure. Figure 7.8 explains the combined result for all subsets.

7.4 YOLO Face Detection and Recognition Results

YOLO face recognition system based on a neural network (DNN). DNN is also a type of convolutional neural network. There are several neural network algorithms used for object detection like Imagenet or single shot detection (SSD). For this research, YOLO was proposed for detection and recognition of faces in low illuminated environments. Figure 6.8 shows the YOLO implementation methodology.

For image classification in R-CNN, the amount of training set is essential in building a model. Illumination on the images plays a vital role in describing the image. As described in Section 5.1, the subset of the images were made on the basis of the angle of light falling on the object.

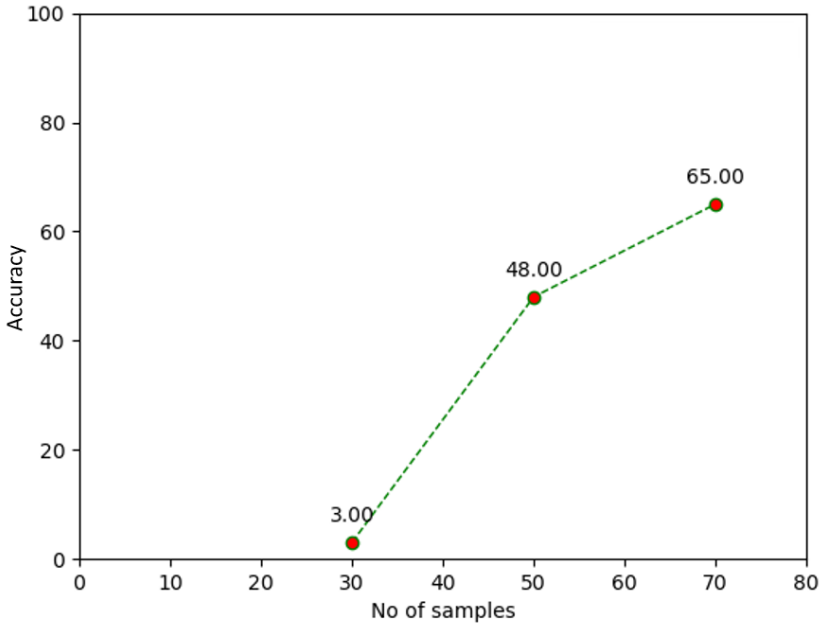


Figure 7.7: Low illumination images with 71° to 130°, Subset 5 result using Classical method(LBPH with KNN)

It is common to use 80% of the data for training and use the remaining 20% for testing. The dataset was divided into five subsets with 647 images per subject. For the training of the network 517 images were used and the remaining 20 percent, i.e 137 images were used for testing and validation of the model. For training purposes, CNN requires large amounts of data, so to increase the dataset size, the data augmentation technique was used, which is explained in chapter 6.2.2.

In our experiments, YOLO was applied on 640x480 size images. From all subjects, two subject images were separated for the validation and accuracy results. Those two subjects are essential to validate the model results. The results for Yolo are shown in table 7.2.

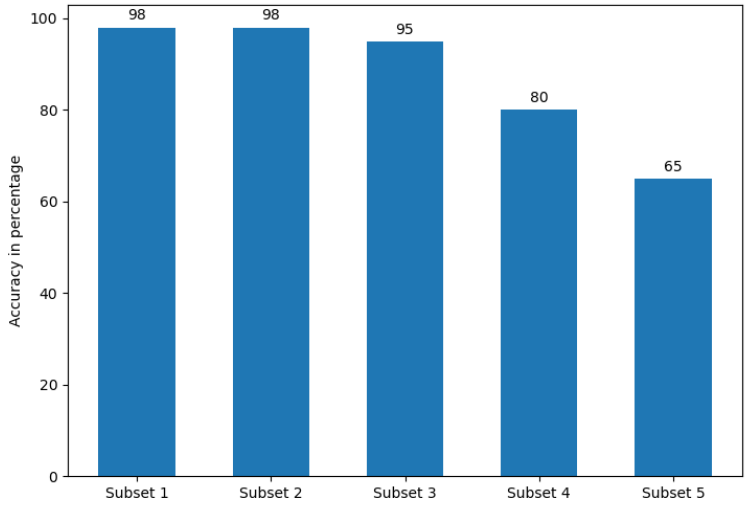


Figure 7.8: Results of LBPH with KNN considering all subsets
Subset 1 is best illumination and subset 5 is represented as low illumination.

Table 7.2: YOLO result on Extended YaleB datset

Total Number of samples	280
False Positive	7.84%
False Negative	4.28%
Overall accuracy	**87.6%**
Recall	**0.92**
Precision	**0.96**

The overall accuracy for the YOLO was found to be 87.6%. However, it is combined result for the five different subsets of the Extended YaleB dataset. Illumination on the images has a very high impact on recognition. Result of face recognition with respect to the subsets is shown in figure 7.9.

From the results of the YOLO algorithm, it can be observed that the illumination on the images affects the accuracy, especially in subset 5. The overall computation complexity of the algorithm remains the same, but the accuracy decreases with the lighting conditions. YOLO showed the results in label percent-

age. Only the high recognition percentage was selected and noticed for the results. For evaluation, the intersection over Union metric was used, the threshold varying from 0.5 to 1.0. At the lower IOU, the detection rate is higher for the images in the low illumination subset five dataset.

Other matrices used for evaluation of detection algorithm are precision and recall. Precision is the ratio of the number of true positives to the total number of positive predictions [103]. The precision of YOLO for all the subset is 0.95 which is higher than previously used algorithm for low illumination. Recall is the ratio of the number of true positives to the total number of actual objects. Recall for YOLO is 0.92 which is acceptable. the recall value for small object in case of YOLO was not good but this is improved because of selection of small anchor boxes in detection process.

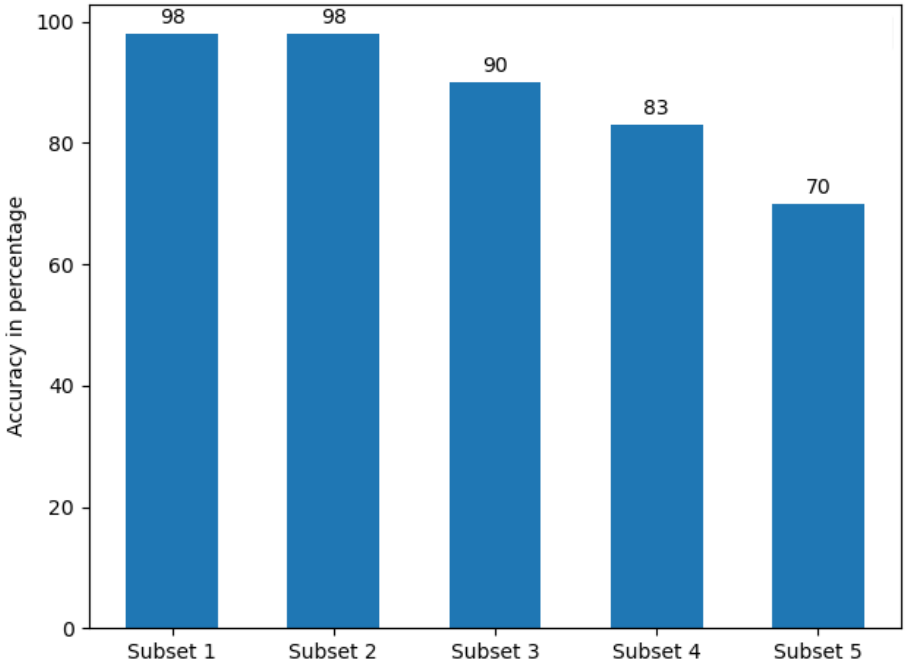


Figure 7.9: Results of YOLO on different subsets

The results of face detection and recognition in a low illuminated environment were improved using low illumination denoising technique which is a popular technique in computer vision [104]. Enhancement of noise is a significant problem

in low illumination visual data. Due to the nature of the mining environment and problem associated with it, preprocessing was performed on the images to improve the quality.

Subset 5 has images with low illumination, which need preprocessing for image enhancement. Image enhancement consists of image quality improvement allowing better visual and computational analysis. For this purpose, Contrast limited adaptive histogram equalization technique (CLAHE) was used [105][106].

7.5 Results for Preprocessed images using CLAHE

CLAHE was designed to improve the performance of adaptive histogram equalization(AHE) [107].In CLAHE *tiles* operates on small area of image. Each tile contrast has been improved such that the output area histogram roughly matches the histogram defined by the "distribution" parameter. The result of CLAHE is shown in figure 7.11.

After applying CLAHE on subset 5 of subject 1, the histogram for image represents the intensity distribution of the image. It represent the pixel for each intensity present in image. 7.10 is the before and 7.11 is after CLAHE, x-axis represents the tonal scale, and y-axis represents the number of pixels in an image. After applying the CLAHE on the subset 5 testing dataset, the results obtained from the YOLO face recognition showed improvement. The comparison of the normal image from subset 5 with the preprocessed image is shown in figure 7.13.

7.6 Critical Analysis

The accuracy of classical LBPH with KNN and YOLO (CNN) was affected by numerous factors, such as pose, size, and illumination. Different illumination of the object produced different results for face detection and recognition. LBPH with KNN results had less accuracy in face detection and recognition in low illumination environments. It is because it detects fewer faces when the subject image had

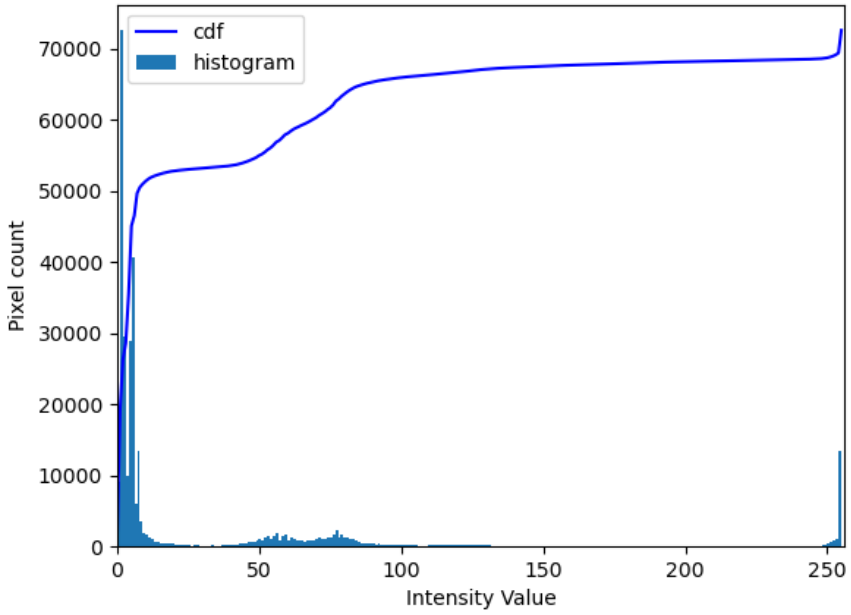


Figure 7.10: Histogram of subset 5, subject 1

low illumination. In low illumination, LBPH with KNN shows false classification error. The overall accuracy of this method is 65 percent. YOLO showed promising results in face detection, but with the standard configuration, YOLO represents the background error which reduces the accuracy. The standard configuration is basically the number of anchor boxes used in training. Large anchor boxes sizes in the standard configuration showed problems in the detection of slim and small faces.

YOLO accuracy was better than LBPH with KNN and it can also be used for real-time face recognition as processing can be done at the rate of 40fps which is real-time in terms of video feed. To reduce the background error in YOLO, anchor boxes were developed which were aligned with the face. These boxes are slim (slim defined as more height than width), this will increase the face detection and lower the background error. Training of YOLO took 72 hours and which is more than that for the classical approach. The training was done on intel i5 without GPU support. The average accuracy for all subsets of YOLO

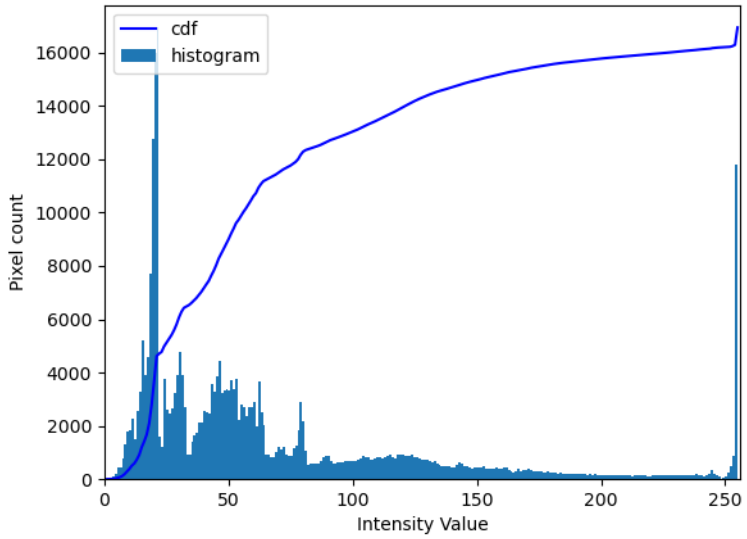


Figure 7.11: Histogram of subset 5, After applying CLAHE on subject 1

was 70 percent, which is better than LBPH with KNN. Table 7.3 explains the accuracy of the algorithm on all subsets.

The detection and recognition of YOLO was improved with the help of preprocessing of images with the "contrast limited adaptive histogram equalization" (CLAHE) approach. CLAHE improves the count of intensity pixels in the low illumination image which helped in the histogram equalization. The average accuracy of YOLO with preprocessed images is 77 percent which is better than

Table 7.3: Accuracy of LBPH with KNN and proposed method

Comparison of LBPH with KNN and YOLO		
Dataset	**LBPH with KNN**	**YOLO**
Subset 1	98	98
Subset 2	98	98
Subset 3	95	90
Subset 4	80	83
Subset 5(Low illumination)	65	70

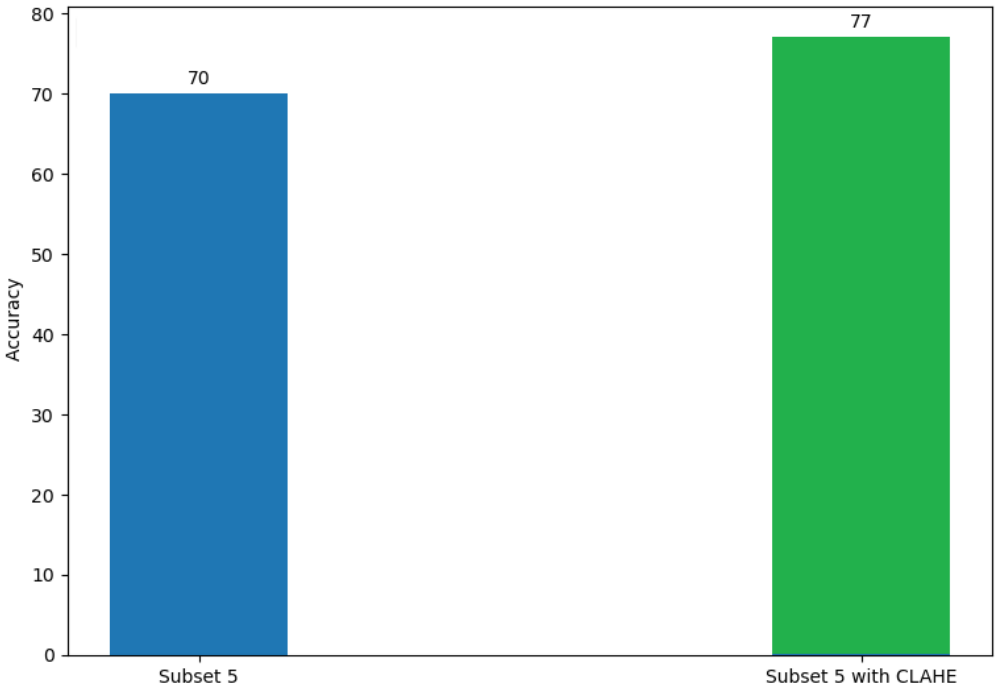


Figure 7.12: Recognition accuracy for subset 5 images, with and without CLAHE.

the classical approach and proposed YOLO based method.

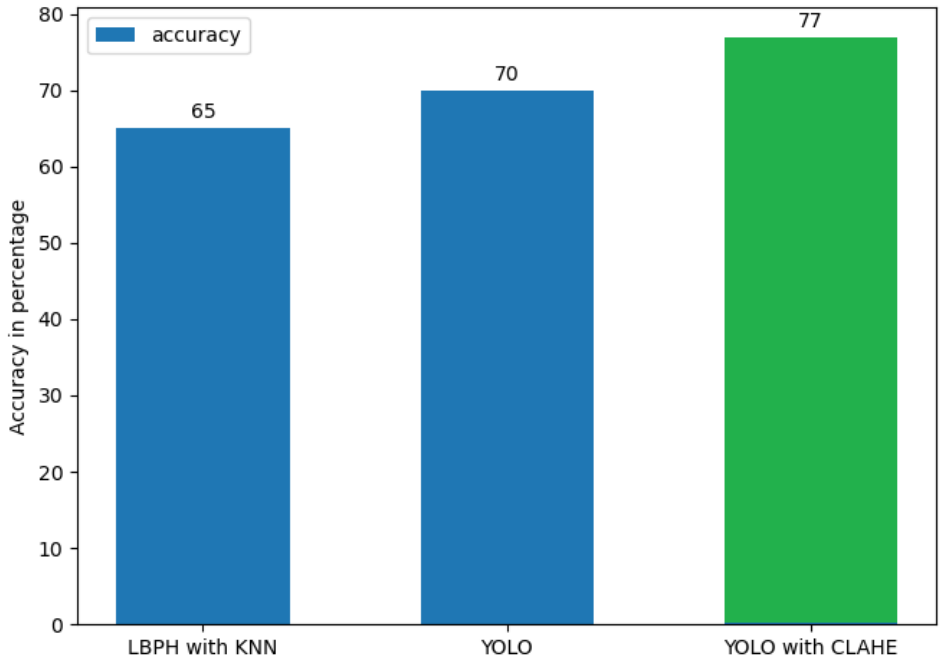


Figure 7.13: Accuracy on comparison of CLAHE and normal subset of low illumination

All the results were calculated using a confusion matrix. It is a specific table that allows visualizing the supervised learning problem. With the help of confusion matrix precision and accuracy were calculated. The results from all the subsets explain that low illumination affects the accuracy of the algorithm. Subset 5 consists of the least lighting condition and the accuracy results for this subset are very low in comparison to all other subsets. For subset 5, LBPH with KNN has 65 percent results and from YOLO, the accuracy is 70 percent. Both of these present less accuracy than CLAHE. Figure 7.14 shows the results for LBPH with KNN, YOLO and the result after processing with CLAHE.

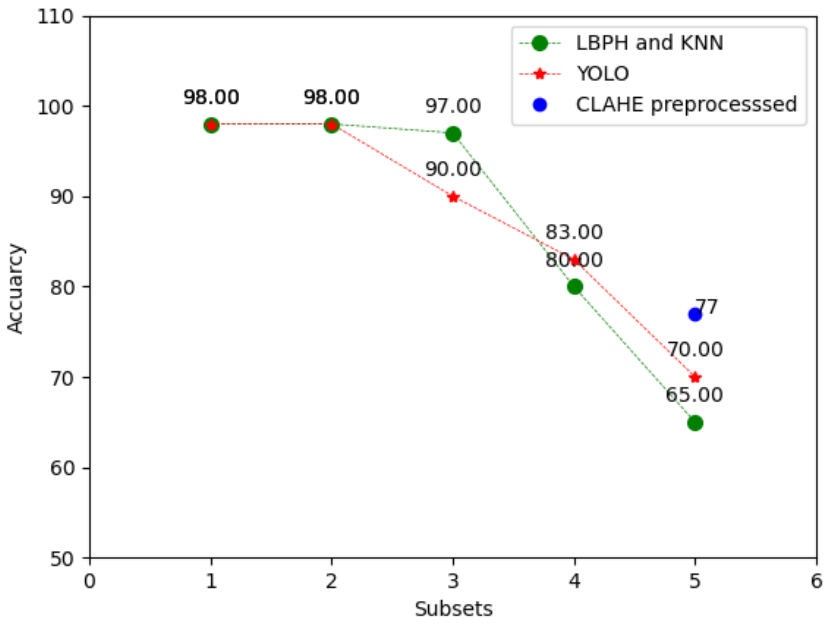


Figure 7.14: Combined results for a)LBPH with KNN 2)YOLO 3) YOLO using Clahe for preprocessing

7.7 Conclusion

The YOLO algorithm for low illumination images showed a better result than that by the LBPH with KNN algorithm. The result also demonstrate the impact of illumination conditions on accuracy. It was observed that the YOLO algorithm takes more time in training as compared to LBPH with KNN. CLAHE (contrast limited adaptive histogram equalization) technique was applied to the testing images and the corresponding results were better than those without the use of CLAHE, as shown in the figure 7.13. This shows that the preprocessing of images improved the accuracy.

Chapter 8

Conclusion recommendation and future work

8.1 Conclusion

The traditional approaches of computer vision are very promising when it comes to face detection and recognition of still images and video sequence in a luminous environment. The dissertation focused on real time video based face detection and recognition in low illumination using LBPH with KNN and YOLO. The goals of this research is:

- Evaluate the results, in terms of accuracy and performance (time complexity) of KNN with LBPH and CNN(YOLO) for face detection and recognition in low light environment (20-40 lux)?
- To find suitable hyperparameters for face detection and recognition in low illumination environments.
- What is the accuracy of performing for face detection and recognition using YOLO for low illumination images?

The first goal indicates the implementation of the classical approach and YOLO for face detection and recognition and evaluate on the basis of accuracy

and performance. The average accuracy across the five illumination subsets for LBPH with KNN is 87.2%. YOLO accuracy for face detection and recognition is 87.6% which is higher than the LBPH with KNN. The luminance of images affect both algorithms. the results for subset 5 (low illuimination dataset) defines the results in figure 7.14. The accuracy for LBPH with KNN and YOLO is 65% and 70% on subset 5. The results of YOLO was improved by applying CLAHE on low illumination images. After applying CLAHE, the accuracy for YOLO increased by 7%.

The second research question was about the fine tuning YOLO. Hyperparameter in the implementation of YOLO play an important role. Hyperparameters were fine tuned to improve the accuracy of the algorithm. The size of the anchor boxes was selected in training and testing with the help of exploratory data analysis.

8.2 Recommendations

The obtained results from CNN support the hypothesis that face detection or recognition using the YOLO algorithm performs better in the low illumination environment. Taking into account these results, CNN based algorithms for real-time face detection and recognition are strong candidate for touch-less access control for different industries. Under current circumstances where the world is facing COVID-19, Face recognition based access controls can helps in creating touch-less access control systems.

8.3 Future work

Face recognition with low illumination shows good results in term of accuracy, which opens new avenues of future work. YOLO shows good and enhanced results when comparing LBPH with KNN but there is room for improvement in it.

The image enhancement techniques used in this research show that the

accuracy of CNN can be improved using image enhancement. Thermal imaging or fused imaging also represent a potential approach to improving face detection and recognition in low illumination environments.

Appendix A

Data Annotation

Data annotation is an important step for data labelling and time taken if the data is big. Precisely annotated data is important and the accuracy is dependent on that quality of data annotation. In training step, annotation will help to detect the image and provide features to the neurons of the network. For annotation of the images, the following code was implemented.

```
import os
import cv2
from lxml import etree
import xml.etree.cElementTree as ET


def write_xml(folder, img, objects, tl, br, savedir):
    if not os.path.isdir(savedir):
        os.mkdir(savedir)
    image = cv2.imread(img.path)
    height, width, depth = image.shape
    annotation = ET.Element('annotation')
    ET.SubElement(annotation, 'folder').text = folder
    ET.SubElement(annotation, 'filename').text = img.name
    ET.SubElement(annotation, 'segmented').text = '0'
    size = ET.SubElement(annotation, 'size')
    ET.SubElement(size, 'width').text = str(width)
```

```
    ET.SubElement(size, 'height').text = str(height)
    ET.SubElement(size, 'depth').text = str(depth)
    for obj, topl, botr in zip(objects, tl, br):
        ob = ET.SubElement(annotation, 'object')
        ET.SubElement(ob, 'name').text = obj
        ET.SubElement(ob, 'pose').text = 'Unspecified'
        ET.SubElement(ob, 'truncated').text = '0'
        ET.SubElement(ob, 'difficult').text = '0'
        bbox = ET.SubElement(ob, 'bndbox')
        ET.SubElement(bbox, 'xmin').text = str(topl[0])
        ET.SubElement(bbox, 'ymin').text = str(topl[1])
        ET.SubElement(bbox, 'xmax').text = str(botr[0])
        ET.SubElement(bbox, 'ymax').text = str(botr[1])
    xml_str = ET.tostring(annotation)
    root = etree.fromstring(xml_str)
    xml_str = etree.tostring(root, pretty_print=True)
    save_path = os.path.join(savedir, img.name.replace('png', 'xml'))
    with open(save_path, 'wb') as temp_xml:
        temp_xml.write(xml_str)
```

Format of data annotation will represents the size and width of the image and explanatory data analysis was run on this data. XML for amt also represents the position of annotated object(face) in the image.

```
<annotation>
  <folder>images</folder>
  <filename>yaleB17_P03A-005E+10.jpg</filename>
  <segmented>0</segmented>
  <size>
    <width>640</width>
    <height>480</height>
    <depth>3</depth>
  </size>
  <object>
```

```
    <name>Subject2</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>280</xmin>
      <ymin>113</ymin>
      <xmax>485</xmax>
      <ymax>398</ymax>
    </bndbox>
  </object>
</annotation>
```

Data augmentation code

```
augmentation_data = ImageDataGenerator(
                    rotation_range=40,
                    width_shift_range=0.2,
                    height_shift_range=0.2,
                    shear_range=0.2,
                    zoom_range=0.2,
                    horizontal_flip=True,
                    fill_mode='nearest')
img = load_img('subset_1.jpg')
x = img_to_array(img)
x = x.reshape((1,) + x.shape)


face_data = 0
for batch in augmentation_data.flow(x, batch_size=1,
                            save_to_dir='preview',
                            save_prefix='subset1',
                            save_format='jpg'):
```

```
face_data += 1
if face_data > 20:
    break
```

Bibliography

[1] J. Liang, J. Wang, Y. Quan, T. Chen, J. Liu, H. Ling, and Y. Xu. "Recurrent exposure generation for low-light face detection." *IEEE Transactions on Multimedia*, 2021.

[2] P. Laytner, C. Ling, and Q. Xiao. "Robust face detection from still images." In *2014 IEEE Symposium on Computational Intelligence in Biometrics and Identity Management (CIBIM)*, pp. 76–80. IEEE, 2014.

[3] S. Sreela and S. M. Idicula. "Action recognition in still images using residual neural network features." *Procedia computer science*, vol. 143, pp. 563–569, 2018.

[4] A. K. Tiwari and S. Lade. "An Exploratory Review of Face Recognition Techniques and Algorithms.", 2014.

[5] Y. Li, G. Tong, H. Gao, Y. Wang, L. Zhang, and H. Chen. "Pano-RSOD: A dataset and benchmark for panoramic road scene object detection." *Electronics*, vol. 8, no. 3, p. 329, 2019.

[6] T. Ahonen, A. Hadid, and M. Pietikainen. "Face description with local binary patterns: Application to face recognition." *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788. 2016.

[8] K. S. Dixit, M. Chadaga, S. Savalgimath, G. R. Rakshith, and M. N. Kumar. "Evaluation and evolution of object detection techniques YOLO and

R-CNN." *International Journal of Recent Technology and Engineering*, vol. 8, no. 2s3, 2019.

[9] D. Mwiti. "Guide to Object Detection." *Heartbeat*, 2019.

[10] "Yale Face Database B." URL http://vision.ucsd.edu/~leekc/ExtYaleDatabase/Yale%20Face%20Database.htm.

[11] K. Crowley. "Gangs live underground for months to feed illegal gold trade." URL https://www.fin24.com/Companies/Mining/Gangs-live-underground-for-months-to-feed-illegal-gold-trade-20150126.

[12] U. Brown. "illegal mining." URL https://www.mineralscouncil.org.za/work/illegal-mining.

[13] D. A. Chowdhry, A. Hussain, M. Z. U. Rehman, F. Ahmad, A. Ahmad, and M. Pervaiz. "Smart security system for sensitive area using face recognition." In *2013 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology (CSUDET)*, pp. 11–14. IEEE, 2013.

[14] CSIR. "The Role of Illumination in Reducing Risk to Health and Safety in South African Gold and Platinum Mines." URL http://researchspace.csir.co.za/dspace/bitstream/handle/10204/1855/GAP804.pdf;jsessionid=D6FFCC1E494BDBD55FE668E9C8C0FC46?sequence=1.

[15] M. F. Rahman, F. Sthevanie, and K. N. Ramadhani. "Face recognition in low lighting conditions using fisherface method and clahe techniques." In *2020 8th International Conference on Information and Communication Technology (ICoICT)*, pp. 1–6. IEEE, 2020.

[16] D. Dechow. "The Fundamentals of Machine Vision." *Viitattu*, vol. 31, p. 2018, 2014.

[17] J. Wang and Z. Li. "Research on face recognition based on CNN." In *IOP Conference Series: Earth and Environmental Science*, vol. 170, p. 032110. IOP Publishing, 2018.

[18] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri. "Face recognition systems: A Survey." *Sensors*, vol. 20, no. 2, p. 342, 2020.

[19] I. Kambi Beli and C. Guo. "Enhancing face identification using local binary patterns and k-nearest neighbors." *Journal of Imaging*, vol. 3, no. 3, p. 37, 2017.

[20] O. Kulkarni, S. Deokar, A. Chaudhari, S. Patankar, and J. Kulkarni. "Real Time Face Recognition Using LBP Features." In *2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1–5. IEEE, 2017.

[21] O. A. R. Salim, R. F. Olanrewaju, and W. A. Balogun. "Class Attendance Management System Using Face Recognition." In *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*, pp. 93–98. IEEE, 2018.

[22] P. J. Phillips, H. Moon, P. Rauss, and S. A. Rizvi. "The FERET evaluation methodology for face-recognition algorithms." In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 137–143. IEEE, 1997.

[23] Ş. Işık. "A comparative evaluation of well-known feature detectors and descriptors." *International Journal of Applied Mathematics Electronics and Computers*, vol. 3, no. 1, pp. 1–6, 2014.

[24] L. Lenc and P. Král. "Automatic face recognition system based on the SIFT features." *Computers & Electrical Engineering*, vol. 46, pp. 256–272, 2015.

[25] R.-J. J. Huang. *Detection strategies for face recognition using learning and evolution*. George Mason University, 1998.

[26] M. Naeem, I. Qureshi, and F. Azam. "FACE RECOGNITION TECHNIQUES AND APPROACHES: A SURVEY." *Science International*, vol. 27, no. 1, 2015.

[27] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[28] F. S. Samaria and A. C. Harter. "Parameterisation of a stochastic model for human face identification." In *Proceedings of 1994 IEEE workshop on applications of computer vision*, pp. 138–142. IEEE, 1994.

[29] P. K. Banerjee and A. K. Datta. "An illumination tolerant class specific 2D subspace based face recognition technique using optimum correlation filter." *Optik-International Journal for Light and Electron Optics*, vol. 124, no. 17, pp. 3173–3179, 2013.

[30] S. U. Sharma and D. J. Shah. "A practical animal detection and collision avoidance system using computer vision technique." *IEEE Access*, vol. 5, pp. 347–358, 2016.

[31] R. Girshick, J. Donahue, T. Darrell, and J. Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587. 2014.

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

[33] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu. "Object detection with deep learning: A review." *IEEE transactions on neural networks and learning systems*, 2019.

[34] G. Guo, H. Wang, Y. Yan, J. Zheng, and B. Li. "A Fast Face Detection Method via Convolutional Neural Network." *Neurocomputing*, 2019.

[35] T. M. Kodinariya. "Hybrid Approach to Face Recognition System using Principle component and Independent component with score based fusion process." *arXiv preprint arXiv:1401.0395*, 2014.

[36] O. Misra and A. Singh. "An approach to face detection and alignment using hough transformation with convolution neural network." In *2016 2nd International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Fall)*, pp. 1–5. IEEE, 2016.

[37] K. Vikram and S. Padmavathi. "Facial parts detection using Viola Jones algorithm." In *2017 4th international conference on advanced computing and communication systems (ICACCS)*, pp. 1–4. IEEE, 2017.

[38] H. A. Rowley, S. Baluja, and T. Kanade. "Neural network-based face detection." *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 1, pp. 23–38, 1998.

[39] M. A. Abuzneid and A. Mahmood. "Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network." *IEEE Access*, vol. 6, pp. 20641–20651, 2018.

[40] I. Dagher and R. Nachar. "Face recognition using IPCA-ICA algorithm." *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 6, pp. 996–1000, 2006.

[41] N. Li, H. Huo, and T. Fang. "Integrating segmentation and classification accuracy for accuracy assessment in object-based image analysis." In *2012 International Conference on Audio, Language and Image Processing*, pp. 599–604. IEEE, 2012.

[42] M. Rizwan, M. Islam, and H. Habib. "Local enhancement for robust face detection in poor SNR images." *Int. J. Comput. Sci. Netw. Secur*, vol. 9, pp. 93–96, 2009.

[43] J. R. Beveridge, D. S. Bolme, B. A. Draper, G. H. Givens, Y. M. Lui, and P. J. Phillips. "Quantifying how lighting and focus affect face recognition performance." In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pp. 74–81. IEEE, 2010.

[44] S. Bhardwaj and A. Mittal. "A survey on various edge detector techniques." *Procedia Technology*, vol. 4, pp. 220–226, 2012.

[45] M. Juneja and P. S. Sandhu. "Performance evaluation of edge detection techniques for images in spatial domain." *International journal of computer theory and Engineering*, vol. 1, no. 5, p. 614, 2009.

[46] B. B. Abad et al. "Proposed Image Pre-processing Techniques for Face Recognition Using OpenCV." In *3rd SPUP International Research Conference*. 2018.

[47] M. A. Boden. *Mind as machine: A history of cognitive science*. Oxford University Press, 2008.

[48] A. Wirdiani, P. Hridayami, A. Widiari, K. Rismawan, P. Candradinata, and I. Jayantha. "Face Identification Based on K-Nearest Neighbor." *Scientific Journal of Informatics*, vol. 6, no. 2, pp. 150–159, 2019.

[49] F. Despotovski, M. Gusev, and V. Zdraveski. "Parallel implementation of k-nearest-neighbors for face recognition." In *2018 26th Telecommunications Forum (TELFOR)*, pp. 1–4. IEEE, 2018.

[50] E. Setiawan and A. Muttaqin. "Implementation of K-Nearest Neighbors face recognition on low-power processor." *Telkomnika*, vol. 13, no. 3, p. 949, 2015.

[51] A. P. Singh, S. S. Manvi, P. Nimbal, and G. K. Shyam. "Face recognition system based on LBPH algorithm." *Ijeat. org*, 2019.

[52] N. Sairamya, L. Susmitha, S. T. George, and M. Subathra. "Hybrid Approach for Classification of Electroencephalographic Signals Using Time–Frequency Images With Wavelets and Texture Features." In *Intelligent Data Analysis for Biomedical Applications*, pp. 253–273. Elsevier, 2019.

[53] T. Ojala, M. Pietikainen, and T. Maenpaa. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns." *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[54] X. Zhao and C. Wei. "A real-time face recognition system based on the improved LBPH algorithm." In *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*, pp. 72–76. IEEE, 2017.

[55] A. Ahmed, J. Guo, F. Ali, F. Deeba, and A. Ahmed. "LBPH based improved face recognition at low resolution." In *2018 international conference on Artificial Intelligence and big data (ICAIBD)*, pp. 144–147. IEEE, 2018.

[56] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection." *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , no. 7, pp. 711–720, 1997.

[57] S. Roychowdhury and M. Emmons. "A survey of the trends in facial and expression recognition databases and methods." *arXiv preprint arXiv:1511.02407*, 2015.

[58] E. I. Abbas, M. E. Safi, and K. S. Rijab. "Face recognition rate using different classifier methods based on PCA." In *2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT)*, pp. 37–40. IEEE, 2017.

[59] F. Lei, X. Liu, Q. Dai, and B. W.-K. Ling. "Shallow convolutional neural network for image classification." *SN Applied Sciences*, vol. 2, no. 1, pp. 1–8, 2020.

[60] A. F. Agarap. "An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification." *arXiv preprint arXiv:1712.03541*, 2017.

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

[62] Y.-Q. Wang. "An analysis of the Viola-Jones face detection algorithm." *Image Processing On Line*, vol. 4, pp. 128–148, 2014.

[63] Y. Adini, Y. Moses, and S. Ullman. "Face recognition: The problem of compensating for changes in illumination direction." *IEEE Transactions on Pattern Analysis & Machine Intelligence*, , no. 7, pp. 721–732, 1997.

[64] F. Schroff, D. Kalenichenko, and J. Philbin. "Facenet: A unified embedding for face recognition and clustering." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 815–823. 2015.

[65] T. Ojala, M. Pietikäinen, and D. Harwood. "A comparative study of texture measures with classification based on featured distributions." *Pattern recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[66] P. Kamencay, M. Benčo, T. Mižd oš, and R. Radil. "A new method for face recognition using convolutional neural network." 2017.

[67] R. Gandhi. "R-CNN, Fast R-CNN, Faster R-CNN, YOLO - Object Detection Algorithms.", Jul 2018. URL `https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e`.

[68] J. Pedoeem and R. Huang. "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers." *arXiv preprint arXiv:1811.05588*, 2018.

[69] C. E. Kim, M. M. D. Oghaz, J. Fajtl, V. Argyriou, and P. Remagnino. "A comparison of embedded deep learning methods for person detection." *arXiv preprint arXiv:1812.03451*, 2018.

[70] Wikipedia contributors. "Luminance — Wikipedia, The Free Encyclopedia." `https://en.wikipedia.org/w/index.php?title=Luminance&oldid=1048058804`, 2021. [Online; accessed 5-October-2021].

[71] E. Silverstein and M. Snyder. "Implementation of facial recognition with Microsoft Kinect v2 sensor for patient verification." *Medical physics*, vol. 44, no. 6, pp. 2391–2399, 2017.

[72] A. Rushworth, C. Talbot, F. Von Glehn, R. Lomas, and R. Franz. "Role of illumination in reducing risk to health and safety in South African gold and platinum mines." 2001.

[73] P. Flanagan. "color FERET Database." 2011.

[74] T. Das. "Machine Learning algorithms for Image Classification of hand digits and face recognition dataset." *Machine Learning*, vol. 4, no. 12, pp. 640–649, 2017.

[75] L. Yu, Q. Cao, and A. Zhao. "Principal patterns of fractional-order differential gradients for face recognition." *Journal of Electronic Imaging*, vol. 24, no. 1, p. 013021, 2015.

[76] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: unified, real-time object detection (2015)." *arXiv preprint arXiv:1506.02640*, 2015.

[77] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi. "Convolutional neural networks: an overview and application in radiology." *Insights into imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[78] C. B. Murthy, M. F. Hashmi, N. D. Bokde, and Z. W. Geem. "Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and Embedded Platforms—A Comprehensive Review." *Applied Sciences*, vol. 10, no. 9, p. 3280, 2020.

[79] "The Extended Yale Face Database B." URL `http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html`.

[80] L. Shamir. "Evaluation of face datasets as tools for assessing the performance of face recognition methods." *International Journal of Computer Vision*, vol. 79, no. 3, p. 225, 2008.

[81] M. Muja and D. G. Lowe. "Fast approximate nearest neighbors with automatic algorithm configuration." *VISAPP (1)*, vol. 2, no. 331-340, p. 2, 2009.

[82] T. Ojala, M. Pietikainen, and D. Harwood. "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions." In *Proceedings of 12th International Conference on Pattern Recognition*, vol. 1, pp. 582–585. IEEE, 1994.

[83] W. Zhao. "Research on the deep learning of the small sample data based on transfer learning." In *AIP Conference Proceedings*, vol. 1864, p. 020018. AIP Publishing LLC, 2017.

[84] S. Ren, K. He, R. Girshick, and J. Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[85] W. J. SAN, M. G. LIM, and J. H. CHUAH. "Efficient Vehicle Recognition and Classification using Convolutional Neural Network." In *2018 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pp. 117–122. IEEE, 2018.

[86] T. Fawcett. "An introduction to ROC analysis." *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[87] C. H. Chan and J. Kittler. "Sparse representation of (multiscale) histograms for face recognition robust to registration and illumination problems." In *2010 IEEE International Conference on Image Processing*, pp. 2441–2444. IEEE, 2010.

[88] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. "From few to many: Illumination cone models for face recognition under variable lighting and pose." *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[89] K.-C. Lee, J. Ho, and D. J. Kriegman. "Acquiring linear subspaces for face recognition under variable lighting." *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[90] J. Brownlee. "How to configure image data augmentation in keras, April 2019." *URL https://machinelearningmastery. com/how-to-configure-image-data-augmentation-whentraining-deep-learning-neural-networks*.

[91] I. H. Sarker. "Machine learning: Algorithms, real-world applications and research directions." *SN Computer Science*, vol. 2, no. 3, pp. 1–21, 2021.

[92] M. S. Menon, A. George, and N. Aswathy. "Implementation of a Multitudinous Face Recognition using YOLO. V3." In *2021 Fourth International Conference on Microelectronics, Signals & Systems (ICMSS)*, pp. 1–6. IEEE, 2021.

[93] N.-D. Nguyen, T. Do, T. D. Ngo, and D.-D. Le. "An evaluation of deep learning methods for small object detection." *Journal of Electrical and Computer Engineering*, vol. 2020, 2020.

[94] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell. "Understanding data augmentation for classification: when to warp?" In *2016 international conference on digital image computing: techniques and applications (DICTA)*, pp. 1–6. IEEE, 2016.

[95] R. Z. Cabada, H. R. Rangel, M. L. B. Estrada, and H. M. C. Lopez. "Hyperparameter optimization in CNN for learning-centered emotion recognition for intelligent tutoring systems." *Soft Computing*, pp. 1–10, 2019.

[96] Y. Le Cun and J. Denker. "Sove Solla, Richard Howard and Lawrence Jockel,"Optimal Brain Damage,"." In *Proceedings of 1989 IEEE Conference on Neural Information Processing Systems—Natural and Synthetic*. 1989.

[97] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang. "YOLO-face: a real-time face detector." *The Visual Computer*, pp. 1–9, 2020.

[98] J. Brownlee. "How to control the stability of training neural networks with the batch size." *Machine Learning Mastery. Tomado de: https://machinelearningmastery. com/how-to-control-the-speed-and-stability-oftraining-neural-networks-with-gradient-descent-batch-size*, 2019.

[99] Y. Bengio. "Practical recommendations for gradient-based training of deep architectures." In *Neural networks: Tricks of the trade*, pp. 437–478. Springer, 2012.

[100] D. Mishkin, N. Sergievskiy, and J. Matas. "Systematic evaluation of convolution neural network advances on the imagenet." *Computer Vision and Image Understanding*, vol. 161, pp. 11–19, 2017.

[101] A. Elmahmudi and H. Ugail. "Deep face recognition using imperfect facial data." *Future Generation Computer Systems*, vol. 99, pp. 213–225, 2019.

[102] M. Sharma and A. Batra. "Analysis of distance measures in content based image retrieval." *Global Journal of Computer Science and Technology*, 2014.

[103] R. Padilla, S. L. Netto, and E. A. da Silva. "A survey on performance metrics for object-detection algorithms." In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 237–242. IEEE, 2020.

[104] L. Li, R. Wang, W. Wang, and W. Gao. "A low-light image enhancement method for both denoising and contrast enlarging." In *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3730–3734. IEEE, 2015.

[105] T. Laksmi, T. Madhu, K. Kavya, and S. E. Basha. "Novel image enhancement technique using CLAHE and wavelet transforms." *International Journal of Scientific Engineering and Technology*, vol. 5, no. 11, pp. 507–511, 2016.

[106] S. M. Pizer. "Contrast-limited adaptive histogram equalization: Speed and effectiveness stephen m. pizer, r. eugene johnston, james p. ericksen, bonnie c. yankaskas, keith e. muller medical image display research group." In *Proceedings of the First Conference on Visualization in Biomedical Computing, Atlanta, Georgia*, vol. 337. 1990.

[107] G. Yadav, S. Maheshwari, and A. Agarwal. "Contrast limited adaptive histogram equalization based enhancement for real time video system." In *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2392–2397. IEEE, 2014.