BIOMETRIC MIRROR - EXPLORING ATTITUDE TOWARDS FACIAL AND OBJECT ANALYSIS

## PROJECT REPORT

### (PHASE- II)

Submitted by

### P. BHARATHI REG. NO: 19TH0411

**R. GAJALAKSHMI REG. NO: 19TH0422**

### S. JAYAPRIYA REG. NO: 19TH0431

**Under the Guidance of**

**Dr. A. MEIAPPANE, M.Tech., Ph.D**

Professor

*in partial fulfillment for the award of the degree of*

### BACHELOR OF TECHNOLOGY

**in**

**DEPARTMENT OF INFORMATION TECHNOLOGY**



## MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

### KALITHEERTHALKUPPAM, PUDUCHERRY- 605 107.

**PONDICHERRY UNIVERSITY INDIA**

**JUNE - 2023**

## MANAKULA VINAYAGAR INSTITUTE OF TECHNOLOGY

## PONDICHERRY UNIVERSITY

**DEPARTMENT OF INFORMATION TECHNOLOGY**

## BONAFIDE CERTIFICATE

This is to certify that the Project work titled **“BIOMETRIC MIRROR EXPLORING ATTITUDE TOWARDS FACIAL AND OBJECT ANALYSIS”** is a bonafide work done by **Ms. P**.**BHARATHI [Reg. No. 19TH0411], Ms. R.GAJALAKSHMI [Reg. No.19TH0422], Ms. S.JAYAPRIYA [Reg. No. 19TH0431**] in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Information Technology by Pondicherry University during the academic year 2022 - 2023.

**HEAD OF THE DEPARTMENT PROJECT GUIDE**

**Dr. P. SIVAKUMAR M.E., Ph.D Dr.A.MEIAPPANE,M.Tech., Ph.D**

**Professor / IT Professor / IT**

*Viva Voce Examination held on*.............................

## INTERNAL EXAMINER EXTERNAL EXAMINER

### DECLARATION

This is to certify that the Report entitled **“BIOMETRIC MIRROR EXPLORING ATTITUDE TOWARDS FACIAL AND OBJECT ANALYSIS”** is the bonafide record of independent work done by **P. BHARATHI, REGISTER NO. 19TH0411, R. GAJALAKSHMI, REGISTER NO. 19TH0422 AND S. JAYAPRIYA, REGISTER NO. 19TH0431** for the award of B.Tech. Degree in **DEPARTMENT OF INFORMATION TECHNOLOGY** under the supervision of **DR. A. MEIAPPANE** Certified further that the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred earlier.

### P. BHARATHI -

1. **R. GAJALAKSHMI -**

### S. JAYAPRIYA -

**HEAD OF THE DEPARTMENT PROJECT GUIDE**

**Dr. P. SIVAKUMAR M.E., Ph.D Dr.A.MEIAPPANE,M.Tech., Ph.D**

**Professor / IT Professor /IT**

## ACKNOWLEDGEMENT

We express my sincere thanks to our Chairman and Managing Director **Shri. M. DHANASEKARAN,** for all his encouragement and moral support. We thank our Vice Chairman **Shri.** S.**V. SUGUMARAN,** and our secretary **Dr. K. NARAYANASAMY** for their constant encouragement.

It gives us great pleasure to convey our deep and sincere thanks to our Principal **Dr. S. MALARKKAN**, for giving constant motivation in succeeding our goal.

We extend our sincere and heartfelt thanks to our Head of Department **Dr. P. SIVAKUMAR,** Professor**,** Department of Information Technology, for providing the right ambience for carrying out his project work.

We would like to thank our guide **Dr. A. MEIAPPANE,** Professor, Department of Information Technology, for providing his valuable guidance and suggestions for our project work. We thank him for the continuous encouragement and the interest shown on us to bring out our project work successfully.

We thank all our department faculty members, non-teaching staffs and my friends of Information technology for helping us to complete the document successfully on time.

We would like to express our eternal gratitude to our parents for the sacrifices they made for educating and preparing us for our future and their everlasting love and support.

We thank the Almighty for blessing us with such wonderful people and for being with us always.

**P.BHARATHI R.GAJALAKSHMI S.JAYAPRIYA**

**ABSTRACT**

Nowadays with the continued development of Artificial Intelligence facial emotion recognition has become more popular. Emotion recognition plays a major role in interaction technology. In interaction technology, the verbal components only play one-third of communication and the non-verbal components play two-thirds of communication. A Facial Emotion Recognition (FER) method is used for detecting facial expressions. Facial expression plays a major role in expressing what a person feels and it expresses inner feelings and his or her mental situation or human perspective. The existing system for this project comprises a FER system using Convolutional Neural Networks (CNN) classifier which suffers from many disadvantages of limited accuracy, limited dataset, and limited flexibility. Deep Neural Network (DNN) through feature learning performs data representation well and has gained many successes in learning and complex problems, many studies have been done on the application of deep neural networks to face recognition and many successes have been achieved. This project aims to identify basic human emotions with a combination of gender classification and age estimation along with the object detection. Facial emotions such as happy, sad, angry, fear, surprise, and neutral emotions are considered basic emotions. We have chosen the main dataset which is namely FER-2013, UTKFace and COCO. The success of this face and object analysis project is evaluated using appropriate evaluation metrics, such as precision, recall, F1 score, and accuracy.

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **BONAFIDE CERTIFICATE** | ***ii*** |
|  | **DECLARATION** | ***iii*** |
|  | **ACKNOWLEDGEMENT** | ***iv*** |
|  | **ABSTRACT** | ***v*** |
|  | **LIST OF FIGURES** | ***ix*** |
|  | **LIST OF TABLES** | ***xi*** |
|  | **LIST OF ABBREVIATIONS** | ***xii*** |
| **1** | **INTRODUCTION** | 1 |
|  | 1.1 OVERVIEW | 1 |
|  | 1.1.1 Problem Identification | 2 |
|  | 1.2 CHALLENGES IN DEEP NEURAL NETWORK | 2 |
|  | 1.3 OBJECTIVE OF THE PROJECT WORK | 3 |
|  | 1.4 NEED/SCOPE FOR THE PROJECT WORK | 4 |
|  | 1.5 ORGANIZATION OF THE CHAPTERS | 5 |
| **2** | **LITERATURE SURVEY** | 6 |
|  | 2.1 SURVEY OF THE RELATED WORKS | 6 |
|  | 2.1.1 Biometric Mirror: Exploring values and Attitudes towards Facial Analysis and Automated Decision making. | 6 |
|  | 2.1.2 Facial Emotion Detection Using Deep Learning | 6 |
|  | 2.1.3 Facial Expression Recognition Based Deep Neural Network | 7 |
|  | 2.1.4 Real Time Facial Expression Recognition Based On Deep Neural  Network | 7 |
|  | 2.1.5 Emotion Recognition using Deep Neural Network with Vectorized Facial Features | 8 |

|  |  |  |
| --- | --- | --- |
|  | 2.2 TECHNIQUES / ALGORITHM | 8 |
|  | 2.2.1 Deep Neural Networks | 8 |
|  | 2.2.2 You Only Look Once | 8 |
|  | 2.3 SUMMARY OF LITERATURE REVIEW | 10 |
| **3** | **SYSTEM REQUIREMENTS** | 13 |
|  | 3.1 HARDWARE REQUIREMENTS | 13 |
|  | 3.2 SOFTWARE REQUIREMENTS | 13 |
|  | 3.3 ABOUT THE SOFTWARE | 14 |
|  | 3.3.1 Visual Studio | 14 |
|  | 3.3.2 Onion Framework | 14 |
| **4** | **EXISTING SYSTEM** | 15 |
|  | 4.1 CONVOLUTIONAL NEURAL NETWORKS OVERVIEW | 15 |
|  | 4.2 CONVOLUTIONAL NEURAL NETWORKS ARCHITECTURE | 15 |
|  | 4.3 DRAWBACK/ISSUES IN CONVOLUTIONAL NEURAL NETWORKS | 16 |
|  | 4.4 PROBLEM STATEMENT | 17 |
| **5** | **PROPOSED SYSTEM** | 18 |
|  | 5.1 OVERVIEW OF DEEP NEURAL NETWORKS | 18 |
|  | 5.2 ARCHITECTURE DIAGRAM OF DEEP NEURAL NETWORKS | 18 |
|  | 5.3 LIST OF MODULES | 19 |
|  | 5.3.1 Dataset Collection | 19 |
|  | 5.3.2 Image Acquisition | 20 |
|  | 5.3.3 Face Detection | 20 |
|  | 5.3.4 Feature Extraction | 21 |
|  | 5.3.5 Deep Neural Network (DNN) Classifier | 22 |
|  | 5.4 DATA SET USED IN THIS PROJECT | 23 |

|  |  |  |
| --- | --- | --- |
|  | 5.4.1 FER-2013 | 23 |
|  | 5.4.2 COCO | 24 |
|  | 5.4.3 UTKFace | 24 |
| **6** | **IMPLEMENTATION** | 26 |
|  | 6.1 DIFFERNTIAL LAYERS IN SYSTEM IMPLEMENTATION | 26 |
|  | 6.1.1 Domain Layer | 26 |
|  | 6.1.2 Application Layer | 27 |
|  | 6.1.3 Infrastructure Layer | 27 |
|  | 6.1.4 Client Layer | 27 |
|  | 6.1.5 Server Layer | 27 |
|  | 6.2 SYSTEM IMPLEMENTATION | 28 |
|  | 6.3 SNAPSHOTS | 29 |
| **7** | **EXPERIMENTAL RESULTS** | 33 |
|  | 7.1 OVERVIEW | 33 |
|  | 7.2 PERFORMANCE ANALYSIS | 33 |
|  | 7.3 RESULTS AND DISCUSSION | 34 |
|  | **7**.3.1 Performance analysis for Face emotions | 34 |
|  | 7.3.2 Performance analysis for Age Prediction | 36 |
|  | 7.3.3 Performance analysis for Gender Prediction | 37 |
|  | 7.3.4 Performance analysis for Object Detection | 38 |
| **8** | **CONCLUSION AND FUTURE**  **ENHANCEMENTS** | 40 |
|  | **REFERENCES** | 41 |
|  | **APPENDIX – I (Source Code)** |  |
|  | **APPENDIX – II (List of Publications)** |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **TITLE** | **PAGE NO** |
| Figure 2.1 | YOLO v3 Diagram | 9 |
| Figure 4.1 | Convolutional Neural Networks Architecture | 15 |
| Figure 5.1 | Deep Neural Network Architecture | 18 |
| Figure 5.2 | Data Collection | 20 |
| Figure 5.3 | Image Acquisition | 20 |
| Figure 5.4 | Fuzzy Down Sampling | 21 |
| Figure 5.5 | Feature Extraction | 21 |
| Figure 5.6 | DNN Classifier | 22 |
| Figure 5.7 | A glimpse of the emotion dataset | 22 |
| Figure 5.8 | Sample images of COCO Dataset | 24 |
| Figure 5.9 | A sample of the UTKFace Dataset | 25 |
| Figure 6.1 | Layers of Onion Architecture | 26 |
| Figure 6.2 | Login Page | 29 |
| Figure 6.3 | API Index | 30 |
| Figure 6.4 | Password Authentication | 30 |
| Figure 6.5 | Database | 31 |
| Figure 6.6 | Final Output | 32 |
| Figure 7.1 | Confusion Matrix for seven emotions | 35 |
| Figure 7.2 | Confusion Matrix Model Accuracy | 35 |
| Figure 7.3 | Confusion Matrix for Age Prediction | 36 |
| Figure 7.4 | Confusion Matrix Model Accuracy | 37 |
| Figure 7.5 | Confusion Matrix for Gender Prediction | 37 |

|  |  |  |
| --- | --- | --- |
| Figure 7.6 | Confusion Matrix Model Accuracy | 38 |
| Figure 7.7 | Confusion Matrix for Object Detection | 39 |
| Figure 7.8 | Confusion Matrix Model Accuracy | 39 |

## LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **TABLE NO** | **TITLE** | **PAGE NO** |
| Table 1.1 | Summary of Literature Review | 10 |
| Table 5.1 | Various techniques used in modules | 22 |
| Table 5.2 | Various Datasets and Predictions | 25 |
| Table 7.1 | Result Analysis of Face Emotions | 34 |
| Table 7.2 | Result Analysis of Age Prediction | 36 |
| Table 7.3 | Result Analysis of Gender Prediction | 37 |
| Table 7.4 | Result Analysis of Object Detection | 38 |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| CNN | Convolutional Neural Networks |
| DNN | Deep Neural Networks |
| ONNX | Open Neural Network Exchange |
| PCA | Principal Component Analysis |
| YOLO | You Only Look Once |

**CHAPTER 1 INTRODUCTION**

Facial expression recognition is a technique done via human beings or computers, which includes finding faces in the scene called a face detection database. It has gained great progress in recent years due to improvements in the design and learning of features and face recognition models. As humans have an exceptional ability to recognize people irrespective of their age, lighting conditions, and varying expressions.

Researchers aim to design an FR system that can match or even surpass the human recognition rate which is nearly 97.5%. The techniques used in the best facial recognition systems may depend on the application of the system.

A face analyzer is software that identifies or confirms a person's identity using their face. It works by identifying and measuring facial features in an image. Facial recognition can identify human faces in images or videos, determine if the face in two images belongs to the same person, or search for a face among a large collection of existing images. Biometric security systems use facial recognition to uniquely identify individuals during user onboarding or logins as well as strengthen user authentication activity.

* 1. **OVERVIEW**
* Face Emotion recognition (FER) system identifies a face by matching it with the facial. In this Project, we explore using Deep Neural Network and device getting to know.
* Object detection algorithm is advanced to analyze a various type behind Objects along with Human Face.
* FER-2013 and COCO dataset is used for train the database. This facial features popularity device is observed to be 98% correct in studying the human emotion.
* Deep Neural Network based design models are broadly utilized in classification task because of their remarkable execution in facial investigation and emotion detection.
* The Deep Neural Network includes Feature extraction which extracts Features corresponding to age, gender, emotion. Furthermore CNN includes Feature classification which classifies facial images into the correct age group, gender and emotion such as happy, sad, angry, neutral.

## 1.1.1 Problem Identification

* The Biometric Mirror scans people faces and uses AI to compare their faces against a database of other faces to produce a personality profile of the scanned person.
* The reports include ratings for the individual responsibility, happiness, aggression, attractiveness, weirdness, and emotional stability.
* The project was created for the purpose of examining the ethics of such systems, not to commercialize it.
* The technology provides opportunities to inter personal characteristics from faces in photos, video recording, and camera feeds.

## CHALLENGES IN DEEP NEURAL NETWORK

Up to now, progress in deep learning has mainly been achieved exploring architectural variants validated on an experimental basis only. Few attempts have been made to understand why and how deep learning obtains such impressive performance. Full understanding of how to choose structural features as well as how to efficiently tune hyper-parameters of models (typically performed through a validation set or a cross-validation approach thanks to extremely expensive, from a computational point of view, procedures) is still far from being a reality.

A specific framework for assessment of unsupervised learning is also needed. Evolving (dynamically adapting structure) type deep learning networks constitute a specific challenge. Another important issue concerns computational efficiency. Currently deep models need a significant amount of computational burden to reach state-of-theart performances on medium/large size datasets and mainly for off-line environments. Nowadays, however, the amount of available data is growing at a rapid pace.

A step forward in this direction is to consider online learning from streams of data. Dealing with a stream of data requires the use of bounded constant memory and almost linear time for learning on single input item. While constant memory may not be a relevant issue for a deep network due to the fact that most of the network architectures are static, i.e. the topology of the network is defined before learning takes place and does not change with time, the constraint on time complexity constitutes a serious challenge.

If we look at the nature of data for future applications of deep learning technologies, it is evident that more and more application domains involve data which can naturally be

represented in structured form, such as sequences (time series, audio and video signals, DNA, etc.), trees (XML documents, parse trees, RNA, etc)

Due to the high combinatorial complexity underpinning structured domains, computationally efficient models to learn relations among structured information at different levels of abstraction are needed. An interesting approach to study could be the development of deep versions of Reservoir Computing models. Incremental approaches provide another research alternative, e.g. exploiting the framework introduced in.

## OBJECTIVE OF THE PROJECT WORK

* The objective of face emotion recognition using Deep Neural Networks (DNN) is to train a model that can accurately recognize human emotions based on facial expressions.
* This is a task in the field of computer vision and is useful for a wide range of applications, including but not limited to, human-computer interaction, marketing research, and psychology research.
* The DNN model is trained using a dataset of labeled facial expressions, which may include a range of emotions such as happiness, sadness, anger, surprise, fear, and disgust.
* The model uses the images of the face as input and learns to recognize patterns that correspond to different emotions. Once trained, the model can be used to classify the emotions of a new image of a face.
* The objective of object detection is to develop an system that can automatically identify and locate objects of interest in an image or video.
* Object detection in DNNs typically involves training a model to classify objects and predict their bounding boxes in an image or video.
* Object detection is a critical task in computer vision and has many real-world applications such as autonomous driving, video surveillance, robotics, and medical imaging.
* Both face emotion recognition and object detection are challenging tasks in computer vision that require sophisticated algorithms and deep learning models. These tasks

involve processing large amounts of image and video data and require the ability to accurately detect and classify complex patterns in the data.

## NEED/SCOPE FOR THE PROJECT WORK

* Security and Surveillance: Face emotion recognition and object detection can be used in security and surveillance applications to detect and classify suspicious activities or people.
* Automotive: Object detection can be used in autonomous vehicles to detect and avoid obstacles, while face emotion recognition can be used to improve driver safety and comfort.
* Healthcare: Face emotion recognition can be used to detect patient distress or pain, while object detection can be used to identify and track medical equipment and supplies.
* Retail: Object detection can be used in retail environments to track inventory and detect shoplifting, while face emotion recognition can be used to analyze customer behavior and improve marketing strategies.
* Entertainment: Face emotion recognition can be used in gaming and virtual reality applications to create more realistic and engaging experiences.
* Human Resources: Face emotion recognition can be used in recruitment and hiring to analyze candidate behavior and identify the best fit for a given role

* 1. **ORGANIZATION OF THE CHAPTERS**

For the proper presentation of the work, the project will be divided into eight(8) chapters.

**In Chapter 1** will cover the introduction which talks about the background of the project, overview of the project, challenges in domain, objective of the project work, scope of the project.

**In Chapter 2** will deal with review of related Literature topics, survey of the related works, techniques or algorithms used in existing literature and overall summary.

**In Chapter 3** will deal with system requirements which includes hardware requirements and software requirements.

**In Chapter 4** will look at the existing system overview, architecture diagram, limitations in existing system and problem statement for the project.

**In Chapter 5** will deal with the proposed system overview, architecture diagram, list of modules, dataset used in this project.

**In Chapter 6** will deal with the system implementation, different layers used in system implementation and performance analysis.

**In Chapter 7** will deal with overview, Results and Discussions and snapshots/output of the project.

**In Chapter 8** will deal with conclusions and future scope of the project

**CHAPTER 2**

**LITERATURE SURVEY**

A literature survey, also known as a literature review, is a thorough analysis and summary of the facial emotion, age, gender, and object detection using DNN on a certain topic or area of study (including books, journals, and academic papers). To identify key concepts, theories, techniques, findings, and knowledge gaps, it requires meticulously reviewing, evaluating, and synthesize pertinent literature.

## SURVEY OF THE RELATED WORKS

### Biometric Mirror: Exploring values and Attitudes towards Facial Analysis and Automated Decision making

**Author:** Niels Wouters , Ryan Kelly

**Year:** 2019

**Paper:** ResearchGate Journal

### DESCRIPTION:

In this paper, we discuss Biometric Mirror, a case study that explored opinions about the ethics of an emerging technology. The interactive application distinguished demographic and psychometric information from people’s facial photos and presented speculative scenarios with potential consequences based on their results. We analyzed the interactions with Biometric Mirror and media reports covering the study. Our findings demonstrate the nature of public opinion about the technology’s possibilities, reliability, and privacy implications. Our study indicates an opportunity for case study-based digital ethics research, and we provide practical guidelines for designing future studies.

* + 1. **Facial Emotion Detection Using Deep Learning Author:** Akriti Jaiswal, A. Krishnama Raju, Suman Deb **Year:** 2020

**Paper**: IEEE Conferernce.

### DESCRIPTION:

Deep learning (DL) based emotion detection gives performance better than traditional methods with image processing. This paper presents the design of an Artificial Intelligence (AI) system capable of emotion detection through facial expressions. It discusses about the procedure of emotion detection, which includes basically three main steps: face detection, features extraction, and emotion classification. This paper proposed a Convolutional Neural Networks (CNN) based deep learning architecture for emotion detection from images. The performance of the proposed method is evaluated using two datasets Facial emotion recognition challenge (FERC-2013) and Japaness female facial emotion (JAFFE). The accuracies achieved with proposed model are 70.14 and 98.65 percentage for FERC- 2013 and JAFFE datasets respectively.

* + 1. **Facial Expression Recognition Based Using Deep Neural Network Author:** Junnan Li and Edmund Y. Lam

**Year:**2015

**Paper:** IEEE Conferernce

### DESCRIPTION:

Develop a technique using deep neural network for human facial expression recognition. Images of human faces are preprocessed with photometric normalization and histogram manipulation to remove illumination variance. Facial features are then extracted by convolving each preprocessed image with 40 Gabor filters. Kernel PCA is applied to features before feeding them into the deep neural network that consists of 1 input layer, 2 hidden layers and a soft max classifier. The deep network is trained using greedy layer-wise strategy. We use the Extended Cohn- Kanade Dataset for training and testing. Recognition tests are performed on six basic expressions (i.e. surprise, fear, disgust, anger, happiness, sadness).

* + 1. **Real Time Facial Expression Recognition Based On Deep Neural Network Author:** T. Ambikadevi Amma, M. R. Sruthy,S. Divya,P. Renuka

**Year**:2019

**Paper:** IJRESM Journal

### DESCRIPTION:

Now a days, emotion recognition plays a major role in interaction technology. In interaction technology the verbal components only play a one third of communication and the non-verbal components plays a two third of communication. A facial emotion recognition (FER) method is used for detecting facial expressions. This paper aims to identify basic human emotions with the combination of gender classification and age estimation. The facial emotions such as happy, sad, angry, fear, surprised, neutral emotions are considered as basic emotions.

* + 1. **Emotion Recognition using Deep Neural Network with Vectorized Facial Features Author:** Guojun Yang, Jordi Saumell y Ortoneda and Jafar Saniie

**Year:**2018

**Paper:** IEEE Conference

### DESCRIPTION:

Emotion reveals valuable information regarding human communications. It is common to use facial expressions to express emotions during a conversation. The vectorized facial feature can be used to build an DNN (Deep Neural Network) for emotion recognition. **Using the proposed vectorized facial feature, the DNN can predict emotions with 84.33% accuracy. Nevertheless, compared with CNNs (Convolutional Neural Network) with similar performance, training such DNN requires less time and data.**

## TECHNIQUES/ALGORITHMS

* + 1. **Deep Neural Networks**

Deep neural networks are ones in which there are multiple hidden layers. Since each hidden layer computes a nonlinear transform of the previous layer, multiple hidden layers have the power to generate much more complex features of the input.

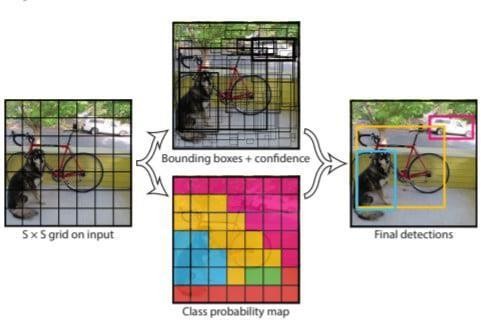
As a result, a deep network can learn significantly more complex functions than a shallow network. It has been shown that a k layer network can represent functions that a (k – 1) layer network can only represent with an exponentially large number of hidden units.

## You Only Look Once Version 3

YOLO (You Only Look Once) v3 is a popular algorithm for object detection in real- time video streams and images. It is based on a deep neural network architecture that predicts object bounding boxes and class probabilities directly from a single image pass. This makes YOLO extremely fast and accurate.

The basic steps involved in using YOLO for object detection are as follows:

* + - 1. Input image: Provide an input image to the YOLO3 algorithm. The image can be of any size, but larger images will take longer to process.
      2. Network architecture: YOLO uses a deep convolutional neural network to predict object bounding boxes and class probabilities. The network architecture consists of multiple convolutional layers followed by fully connected layers.
      3. Preprocessing: YOLO requires the input image to be preprocessed by resizing it to a fixed size and normalizing the pixel values. This ensures that the input data is consistent and the network can learn useful features.
      4. Object detection: YOLO predicts object bounding boxes and class probabilities directly from the input image. It does this by dividing the input image into a grid of cells and predicting a bounding box and class probability for each cell.
      5. Non-maximum suppression: YOLO predicts multiple bounding boxes for each object in the image, so non-maximum suppression is used to remove overlapping boxes and retain the most confident detection.
      6. Output: YOLO outputs a set of bounding boxes and corresponding class probabilities for each detected object in the input image.



**Figure 2.1 YOLO v3 Diagram**

## 2.3 SUMMARY OF LITERATURE REVIEW

**Table 2.1 Summary of literature review**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.No** | **Title of the paper** | **Description** | **Issues/ Drawback/**  **Limitations** |
| 1 | Biometric Mirror: Exploring values and Attitudes towards Facial Analysis and Automated Decision making.  **Author:**Niels Wouters , Ryan Kelly.  **Year:** 2019 /IEEE | The interactive application distinguished demographic and psychometric information from people’s facial photos and presented speculative scenarios with potential consequences based on their results | The main problem with using facial recognition biometrics to login is that the user needs to remove their face mask. Many people don't like the idea o having to remove face masks to access systems in a place that likely requires the use of face masks. |
| 2 | Facial Emotion Detection Using Deep Learning  **Author:**Akriti Jaiswal,  A. Krishnama Raju, Suman Deb **Year:**2020/IEEE | Deep learning (DL) based emotion detection gives performance better than traditional methods with image processing. This paper presents the design of an artificial intelligence (AI) system capable of emotion detection through facial expressions. | Mimics could be to some extent controlled by humans and therefore the recognition results might be intentionally or unintentionally  falsified (Landowska and Miler,2016).  Disadvantages of face detection include huge storage requirements, vulnerable detection, and potential privacy issues. |

|  |  |  |  |
| --- | --- | --- | --- |
| 3 | Facial Expression Recognition Based Using Deep Neural Network  **Author:** Junnan Li and Edmund Y. Lam **Year**:2015/IEEE | Develop a technique using deep neural network for human facial expression recognition. Images of human faces are  preprocessed with photometric normalization and histogram  manipulation to remove illumination variance. | The disadvantages of using facial expressions to measure emotions are that most facial expression coding schemes rely on the FACS system traditionally used to classify only the six basic emotions, and are very labor-intensive if done by trained human coders rather than software |
| 4 | Real Time Facial Expression Recognition Based On Deep Neural Network  **Author:**  T. Ambikadevi Amma,  M. R. Sruthy,S. Divya,P. Renuka **Year**:2019/Journal | This paper aims to identify basic human emotions with the combination of gender classification and age estimation. The facial emotions such as happy, sad, angry, fear, surprised, neutral emotions are considered as basic emotions. | Poor Image Quality. The effectiveness of facial- recognition algorithms is influenced by the image quality. Small Image Sizes. Different Face Angles. Data Processing and Storage Issues. |

|  |  |  |  |
| --- | --- | --- | --- |
| 5 | Emotion Recognition using Deep Neural Network with  Vectorized Facial Features  **Author:** Guojun Yang, Jordi Saumell y Ortoneda and Jafar Saniie  **Year:**2018  **Paper:**IEEE Conference | Emotion reveals valuable information regarding human communications. It is common to use facial expressions to express emotions during a conversation. The vectorized facial feature can be used to build an DNN (Deep Neural Network) for emotion recognition. Using the proposed vectorized facial feature, the DNN can predict emotions with 84.33% accuracy. Nevertheless, compared with CNNs (Convolutional Neural Network) with similar performance, training such DNN requires less time and data. | It is a challenge to make emotion available in different languages.  Performance and results of the emotion sensing system depends on accuracy of the sensors such as cameras, thermal image sensors, facial recognition algorithm used and so on. Highly accurate system will be expensive due to use of costly components. |

**CHAPTER 3**

**SYSTEM REQUIREMENTS**

* 1. **HARDWARE REQUIREMENTS**
     + Processor : Intel Pentium Dual Core 1.8GHz
     + System : Pentium Dual Core.
     + Hard Disk : 120 GB.
     + Monitor : 15’’ LED
     + Input Devices : Keyboard, Mouse
     + Ram : 8 GB
  2. **SOFTWARE REQUIREMENTS**
     + Operating System : Windows 10/Windows 11
     + Coding Language : C#
     + Tool : Visual Studio 2022
     + Database : MySQL
     + Framework : Onion, .NET

## ABOUT SOFTWARE

* + 1. **Visual Studio**

An integrated development environment (IDE) is a feature-rich program that supports many aspects of software development. The Visual Studio IDE is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and many more features to enhance the software development process.

Visual Studio is available for Windows and Mac. [Visual Studio for Mac](https://learn.microsoft.com/en-us/visualstudio/mac/) has many of the same features as Visual Studio for Windows, and is optimized for developing cross- platform and mobile apps. This article focuses on the Windows version of Visual Studio. There are three editions of Visual Studio: Community, Professional, and Enterprise.

With all of these Visual Studio also serves as a testing platform as well. Through this developer can easily test their application how their application is working on target environment and ensure that they do so smoothly once they are deployed.

## Onion Framework

Onion architecture is **built on a domain model in which layers are connected through interfaces**. The idea is to keep external dependencies as far outward as possible where domain entities and business rules form the core part of the architecture. It provides flexible, sustainable and portable architecture.

The research onion suggests **mono-method, mixed method and multi-method** as possible choices for conducting research. The mono-method comprises only one method for the study. The mixed method is based on the use of two or more methods of research and commonly refers to the use of qualitative and quantitative methodology.

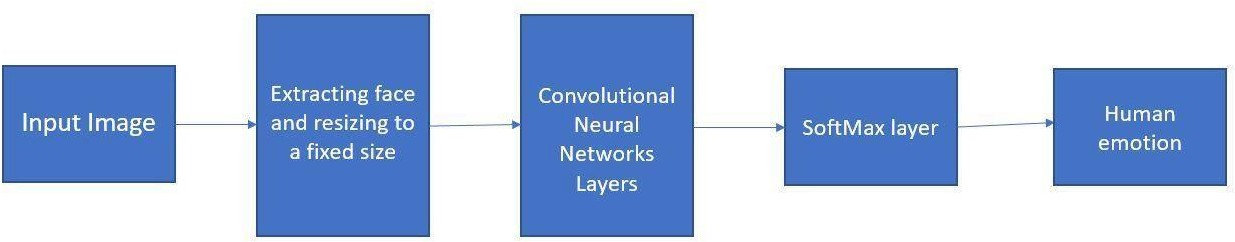
# **CHAPTER 4**

# **EXISTING SYSTEM**

## CONVOLUTIONAL NEURAL NETWORK OVERVIEW

A Convolutional neural network (CNN) is a type of artificial neural network that has one or more convolution layers and are used mainly for image processing, classification, segmentation and also for other auto correlated data. Deep learning is a machine learning based artificial neural network that recognize objects in image by progressively extracting features from data through higher layers. As shown in figure in order to recognize face in an image we have to train the CNN with human faces. The benefit of using CNNs is their ability to develop an internal representation of a two dimensional image. This allows the model to learn position and scale of faces in an image. After train the CNN it can able to recognize face in an image One can effectively use Convolutional Neural Network for Image data. CNN that extracts features in an image.

## CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE



**Figure 4.1 Convolutional Neural Network Architecture**

Here is a high-level overview of a typical architecture:

* + - Input: The system takes an image or a video stream as input, which may be preprocessed to enhance image quality and remove noise.
    - Feature extraction: The input is fed through one or more convolutional layers to extract features from the image. These layers apply filters to the input, which help to identify edges, shapes, and patterns in the image.
    - Pooling: The output of the convolutional layers is typically down sampled using pooling layers, which reduce the dimensionality of the feature maps and help to avoid over fitting.
    - Soft max Layer: The use of the soft max layer in CNN-based face emotion recognition allows for probabilistic predictions, which can be useful in cases where there is uncertainty or ambiguity in the input. It also enables the use of standard classification metrics such as cross-entropy loss, which can be used to train the model to minimize prediction error.
    - Classification: The resulting feature maps are then flattened and fed into one or more fully connected layers for classification. These layers use the extracted features to classify the image as belonging to a particular emotion or object class.
    - Output: The system produces an output, which may include the class label or probability scores for each class.

## DRAWBACK/ISSUES IN CONVOLUTIONAL NEURAL NETWORK

* + - Though much progress has been made recognizing facial expression with a high accuracy remains difficult due to the subtlety, complexity and variability of facial expressions.
    - Low-resolution images in real world environments make real-life expression recognition much more difficult.
    - And we must also consider the factor of time and memory.

## PROBLEM STATEMENT

The existing system for this project comprises facial emotion recognition using Convolutional Neural Networks which suffers from the following issues:

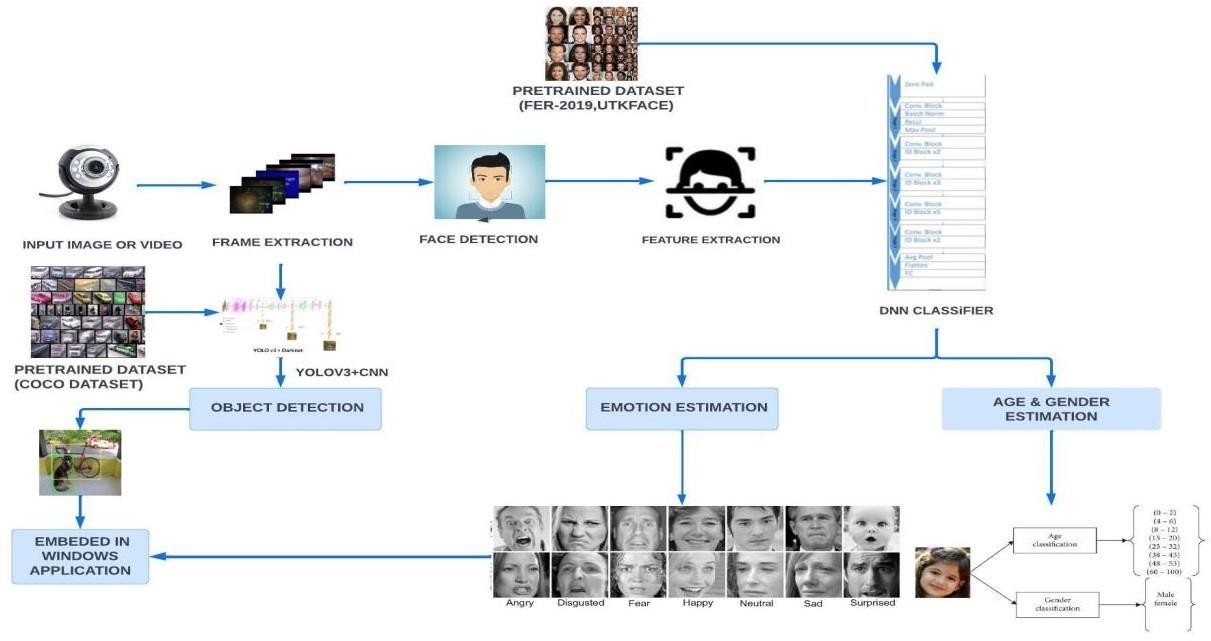
* + - Limited accuracy: Although CNNs can achieve high accuracy in face emotion recognition and object detection, there may still be errors in the predictions. This can be due to several factors, such as variations in lighting, pose, and occlusions, as well as the complexity and variability of emotions.
    - Need for large datasets: CNNs require large, diverse datasets for training. Collecting and labeling large datasets for face emotion recognition and object detection can be time-consuming and expensive.
    - Limited flexibility: CNNs are designed to work with fixed-size inputs and are less flexible in terms of handling images with different sizes or aspect ratios. This may limit the applicability of the model to real-world scenarios where input images can vary widely in terms of size and aspect ratio.

To overcome the issues, we build a face emotion recognition with the help of Deep Neural Networks (DNN) which results the output with the maximum accuracy and which is done with less time and data.

**CHAPTER 5**

**PROPOSED SYSTEM**

* 1. **OVERVIEW OF DEEP NEURAL NETWORKS**
     + The proposed system “Biometric mirror” has two different detection like face emotion detection and object detection. These detection finds the face emotion and object in real time with the help of an algorithm such as Deep Neural Networks and YOLO.
     + Facial features will be introduced to represent facial expression. The proposed facial feature model can not only reflect facial expressions correctly, it can also be used for DNN with high efficiency. To test the efficiency of such method, a DNN is trained to recognize some universal expressions.
     + Compared with more complicated neural network structures, such as CNNs (Convolutional Neural Network), a DNN can be trained with less data, therefore, to be built quicker. To be more specific, with simpler inputs, a deep neural network can have a simpler structure, hence, can be trained with less data, and quicker.
  2. **ARCHITECTURE DIAGRAM FOR DEEP NEURAL NETWORKS**

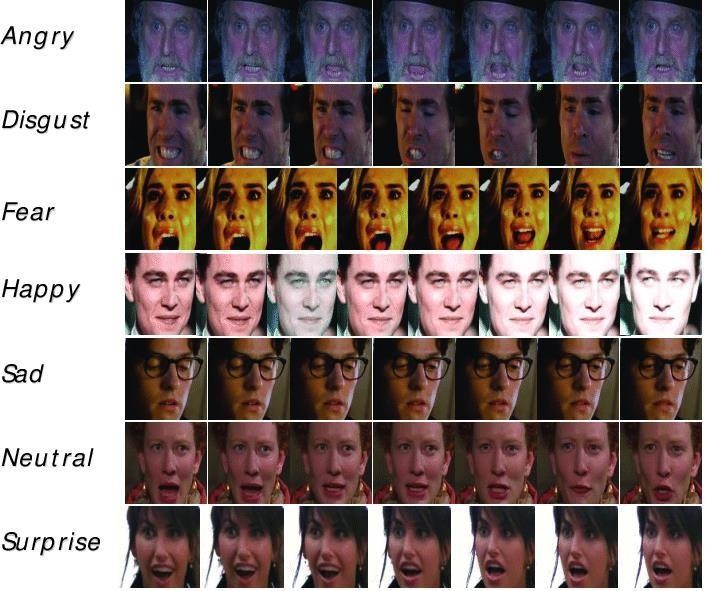


**Figure 5.1 Deep Neural Network Architecture**

* + - Firstly, we collect the dataset of COCO and FER-2013 in the process of Face emotion recognition and object detection.
    - Then it compare with input image or video, it produce frame capturing.
    - Then frame capture analyze face detection using viola Jones Algorithm with having Face Feature Extractions to face datasets.
    - It will extracts the select facial feature using Ada boost Algorithm.
    - Next, we select the some facial features for DNN classifier, it will detect the face emotions with some characteristics (Emotion, Age & gender and Object).
    - Finally output is embedded as windows application.
  1. **LIST OF MODULES**

The list of modules in the proposed system includes image acquisition, 2d to 3d Conversion, fuzzy down sampling, feature extraction, feature selection and deep neural networks.

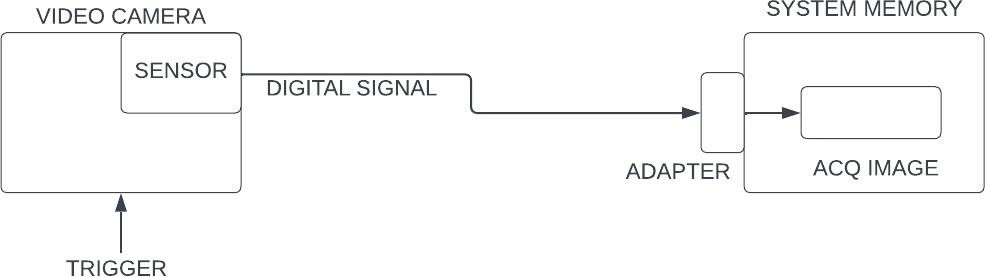
* + 1. **Dataset Collection**
       - The dataset used in this research is Facial Emotion Recognition 2013 (FER2013). The training set consists of 28,709 examples, and the testing set consists of 7179 examples. The data consist of 48 x 48 pixel grayscale images of the face. On the other hand, the test data set consists of 70 examples.
       - The COCO (Common Objects in Context) dataset is a large-scale [image](https://www.v7labs.com/blog/image-recognition-guide) [recognition](https://www.v7labs.com/blog/image-recognition-guide) dataset for object detection, segmentation, and captioning tasks. It contains over 330,000 images, each annotated with 80 object categoriesand 5 captionsdescribing the scene.



### Figure 5.2 Data Collection

## Image Acquisition

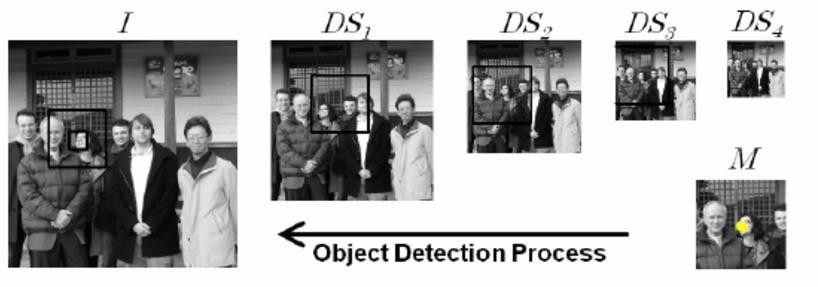
* + - * Before any video or image processing can commence an image must be captured by a camera and converted into a manageable entity.
      * This is the process known as image acquisition.



**Figure 5.3 Image Acquisition**

## 5.3.3 Face Detection

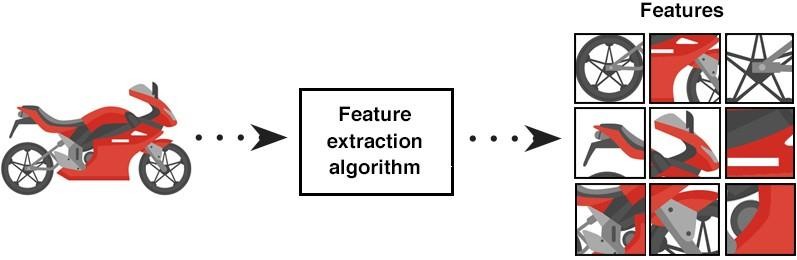
* + - * Face Detection component was implemented by Viola Jones.
      * Image is rescaled to 64 \* 64 px by Fuzzy Down Sampling.



**Figure 5.4 Fuzzy down Sampling**

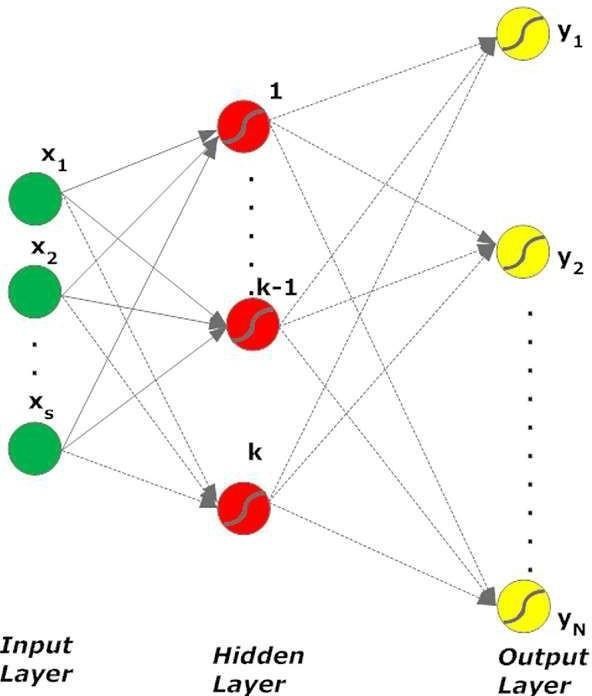
## 5.3.4 Feature Extraction

* + - * Feature extraction is a special form of dimensionality reduction.
      * Extract features from the Face through the evaluation of their weights in different related domains. The feature extraction is a preprocessing stage of the knowledge discovery.
      * This preprocessing step aims at converting the face review features into a set of specific features and, at the same time, enriching their semantic characteristics.



**Figure 5.5 Feature Extraction**

## Deep Neural Network (DNN) Classifier



### Figure 5.6 DNN Classifier

* Deep Neural Network (DNN) has recently achieved outstanding performance in a variety of computer vision tasks, including facial attribute classification.
* The great success of classifying facial attributes with DNN often relies on a massive amount of labelled data. However, in real-world applications, labelled data are only provided for some commonly used attributes (such as age, gender); whereas, unlabelled data are available for other attributes (such as attraction, responsibility).

## Table 5.1 Various Techniques Used In Modules

|  |  |  |
| --- | --- | --- |
| **S.No** | **Modules** | **Techniques Used** |
| 1 | Face Detection | Viola Jones |
| 2 | Feature Extraction | Darknet |
| 3 | Feature Selection | PCA |
| 4 | Classification | DNN |

* 1. **DATA SET USED IN THIS PROJECT**
     1. **FER-2013 Dataset**

There are multiple international datasets utilized in different research works on the topic of emotion recognition. We have used one such dataset known as the FER – 2013 dataset which can be found on the Kaggle repository (<https://www.kaggle.com/datasets/msambare/fer2013>). The images in this dataset have been clustered into 8 different directories, which are:

* + - * Neutral
      * Happiness
      * Surprise
      * Sadness
      * Anger
      * Disgust
      * Fear
      * Contempt

**Figure 5.7 A glimpse of the emotion dataset**

Each of these categories consists of nearly 1000 images, each being 48 (x) 48 size and formatted in. “.png” format. These faces were automatically recorded such that they are almost cantered in each picture and take up around the same amount of area.

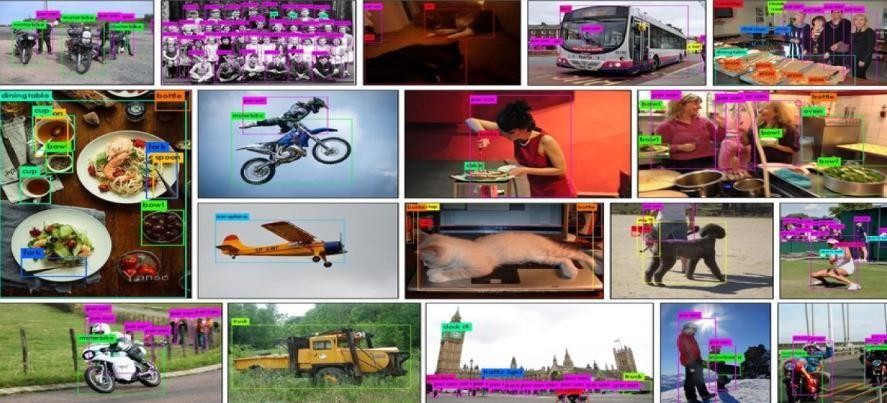
## COCO Dataset

The COCO (Common Objects in Context) dataset is a large-scale [image](https://www.v7labs.com/blog/image-recognition-guide) [recognition](https://www.v7labs.com/blog/image-recognition-guide) dataset for object detection, segmentation, and captioning tasks. It contains over 330,000 images, each annotated with 80 object categories and 5 captions describing the scene. The COCO dataset is widely used in computer vision research and has been used to train and evaluate many state-of-the-art object detection and segmentation models.

https://pjreddie.com/media/files/yolov3.weights https://github.com/pjreddie/darknet/blob/master/cfg/yolov3.cfg https://github.com/pjreddie/darknet/blob/master/data/coco.names

[Object detection](https://www.v7labs.com/blog/object-detection-guide) is the most popular computer vision application. It detects objects

with bounding boxes to enable their classification and localization in an image.

The COCO dataset can be used to train object detection models. The dataset provides [bounding box](https://www.v7labs.com/blog/bounding-box-annotation) coordinates for 80 different types of objects, which can be used to train models to detect bounding boxes and classify objects in the images.

**Figure 5.8 Sample images of COCO Dataset**

## 5.4.3 UTKFace Dataset

For the age and gender identification part of the model, we have considered the UTKFace dataset which can be accessed through this web link – (https://susanqq.github.io/UTKFace/). It is a huge substantial facial dataset with a wide age stretching from 0 to 116 age groups of human faces. The compilation of image data consists of more than 20,000 facial photographs with classifications for age groups, ethnic groups, and gender. The images feature a variety of stances, facial gestures, and expressions, conditions illumination, backlighting, clarity, etc.



### Figure 5.9 A sample of the UTKFace dataset

**Table 5.2 Various Datasets and Predictions**

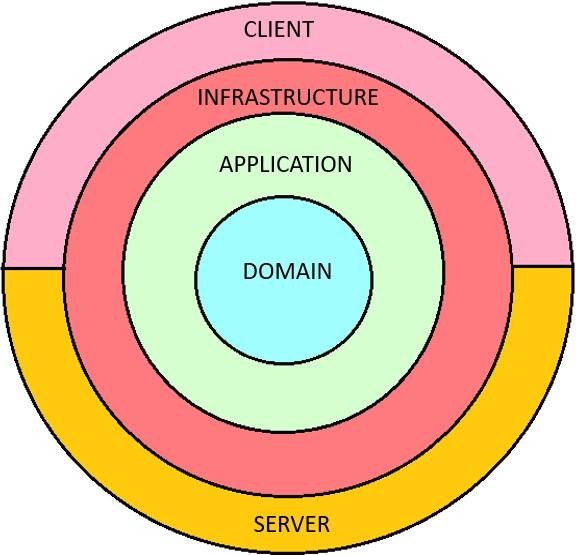
|  |  |  |
| --- | --- | --- |
| **S.No** | **Dataset** | **Features Prediction** |
| 1 | FER-2013 | Face Emotions |
| 2 | COCO | Object Detection |
| 3 | UTKFACE | Age, Gender Prediction |

**CHAPTER 6**

**IMPLEMENTATION**

Implementation is the execution or practice of a plan, a method or any design, idea, model, specification, standard or policy for doing something. As such, implementation is the action that must follow any preliminary thinking for something to actually happen. All the modules in the theoretical design have been implemented in the following phases.

* 1. **DIFFERENT LAYERS USED IN SYSTEM IMPLEMENTATION**



**Figure 6.1 Layers of Onion Architecture**

* + 1. **DOMAIN LAYER**

This layer lies in the center of the architecture where we have application entities which are the application model classes or database model classes. Using the code first approach in the application development using Asp.net core these entities are used to create the tables in the database. It contains Dependencies where analysers and frameworks are used. Various entities such as Auditable and prediction logs are used.

* + 1. **APPLICATION LAYER**

This layer comes next to the domain layer. In this it has folders such as configurations, enums, exceptions, extensions, features, interfaces, models, requests, responses and validators. Application Layer contains overall structure of the app where it uses validators to check the users who login. App and Mail configurations are used in this modules. Basic Commands and Queries are used for the MySQL user.

* + 1. **INFRASTRUCTURE LAYER**

This layer lies next to the application layer. In this it has folders such as configurations, enums, exceptions, extensions, features, interfaces, models, requests, responses and validators. Infrastructure layer consists the permissions, models, Database seeder, storage process. It constructs the overall structure of the project so that it can work well along with other layers.

* + 1. **CLIENT LAYER**

In our project, the client layer contains the user interface part where it contains Authentication, Models, properties, Dependencies, Communication, custom icons, identity and resources. The client layer contains the UI part which the user can see in the frontend part. This is named as Maui Program.cs so that it can be run in the windows machine.

* + 1. **SERVER LAYER**

The server layer is that connects to the client layer where server is backend part which helps to run the client layer correctly. It also contains Authentication, Models, properties, Dependencies, Communication, custom icons, identity and resources. The server is connected with the MySQL workplace to add or remove users with the help of administrators. Server is that manages the user roles, user details and login details. The server layer acts as the backbone of an application, handling authentication, managing user details and roles, interacting with databases, and facilitating communication between clients and other system components. It's important to note that the actual implementation of the server layer for facial and object detection can vary depending on the specific application or system requirements.

## SYSTEM IMPLEMENTATION

**Phase 1:** Designing the Architecture of the application

The project is designed using the onion architecture in order to build the application on the domain model where the layers are connected through the interfaces. It is designed to make the app flexible, sustainable and portable.

**Phase 2:** Data Collection

The data is collected and stored in the database using the domain module where all the features of the database is coded there. Domain entity layer is the center part of the architecture. It holds all application domain objects.

**Phase 3:** Developing the Application outlook

The overall outlook of the application is done in **BiometricMirror.Application**. The implementation of the interfaces is done using services and request objects in the application module.

**Phase 4:** Implementation using Infrastructure

The services and response interaction between the server and client is coded using the infrastructure module.

**Phase 5:** User Interface Infrastructure

The client interaction with the server is taken care by the UI module which provides the root authentication of the client. The interaction platform for the client with the server is implemented using this module.

**Phase 6:** The Application Programming Interface(API)

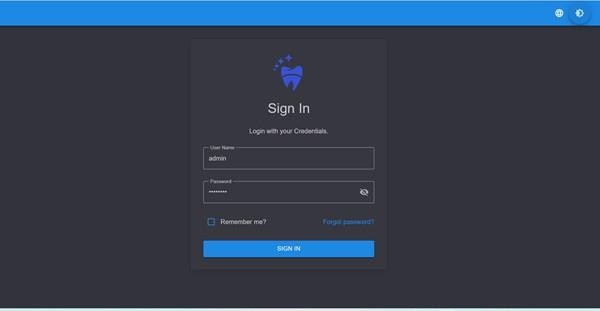
An API (Application Programming Interface) is a set of rules, protocols, and tools that allows different software applications to communicate and interact with each other. It defines the methods, data formats, and authentication mechanisms that enable one software application to access and utilize the features or services of another application, system, or platform. **BiometricMirror.API** used to connect client and server process. This comes under server module.

**Phase 7:** MAUI

NET Multi-platform App UI (.NET MAUI) is a cross-platform framework for creating native mobile and desktop apps with C# and XAML. Using .NET MAUI, you can develop apps that can run on Android, iOS, macOS, and Windows from a single shared code-base. It comes under the client Module.

* 1. **SNAPSHOTS STEP-1** LOGIN PAGE

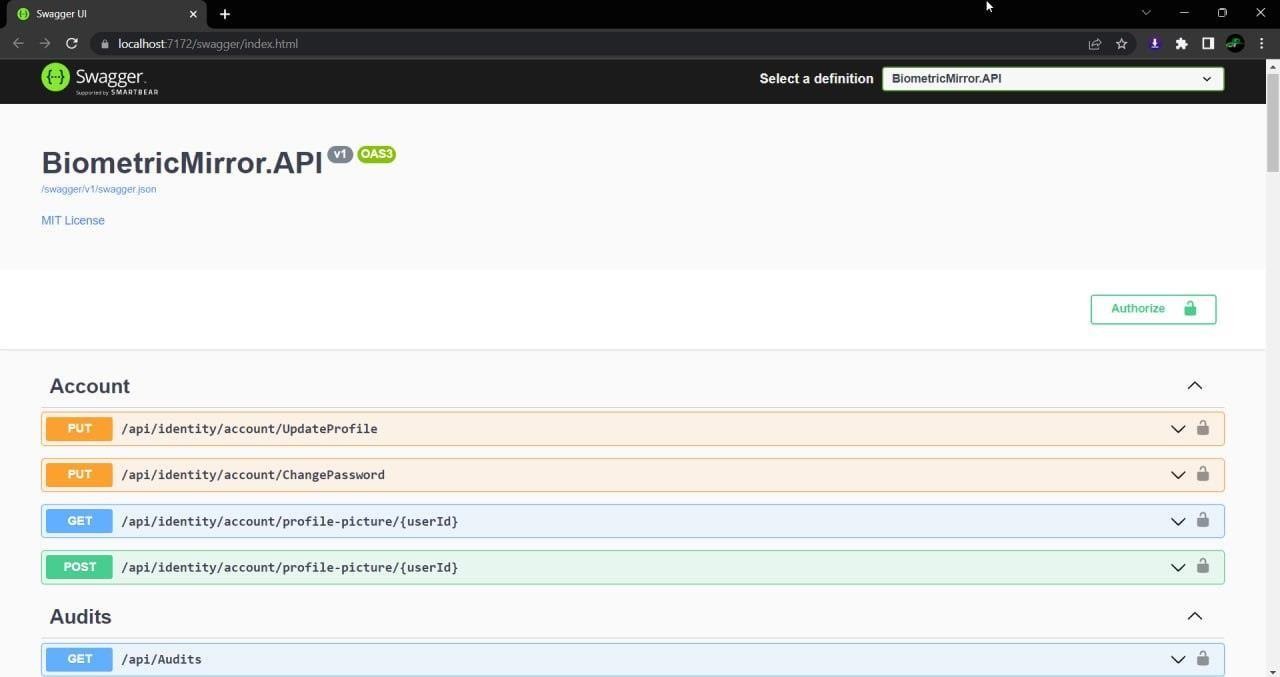
A login page is a user interface component that allows users to authenticate themselves and gain access to a restricted area or functionality within the application. It is typically the first interaction point for users to provide their credentials (such as username and password) and verify their identity.



**Figure 6.2 Login Page**

**STEP-2** API INDEX

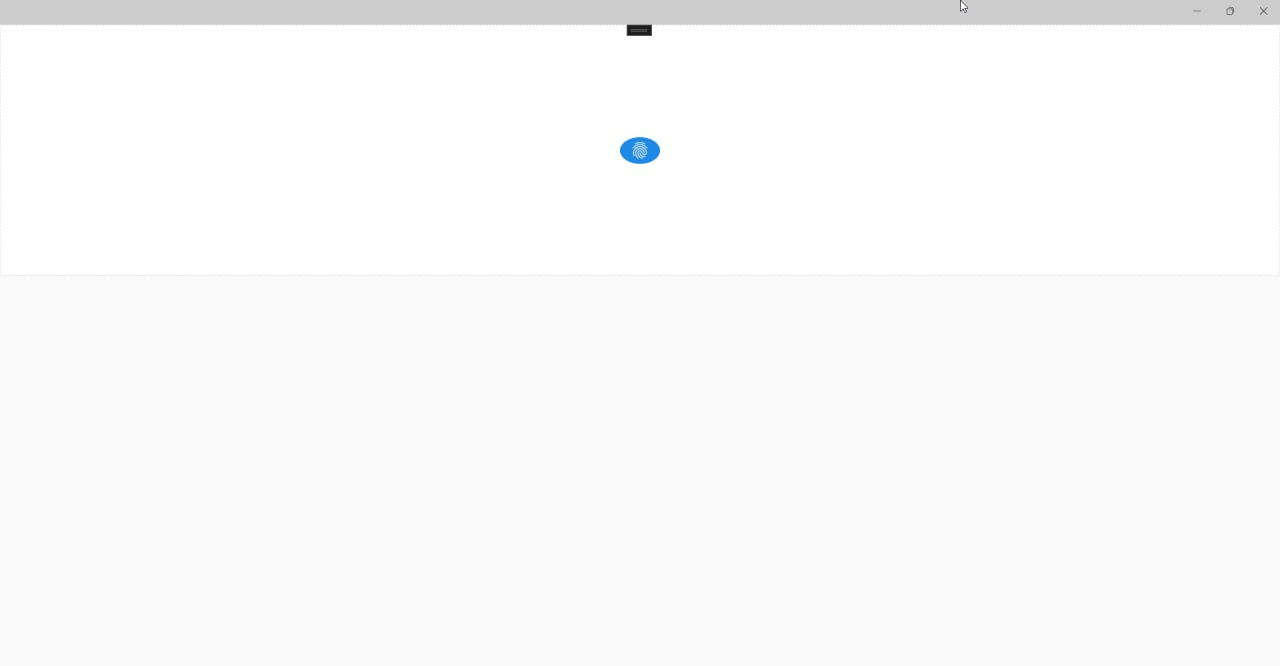
API index refers to a central documentation or reference page that provides an overview and details of the available APIs (application programming interfaces) within the project. It serves as a comprehensive guide for developers to understand the functionality, endpoints, parameters, and response formats of the APIs offered by the software project.



**Figure 6.3 API Index**

**STEP-3** PASSWORD AUTHENTICATION

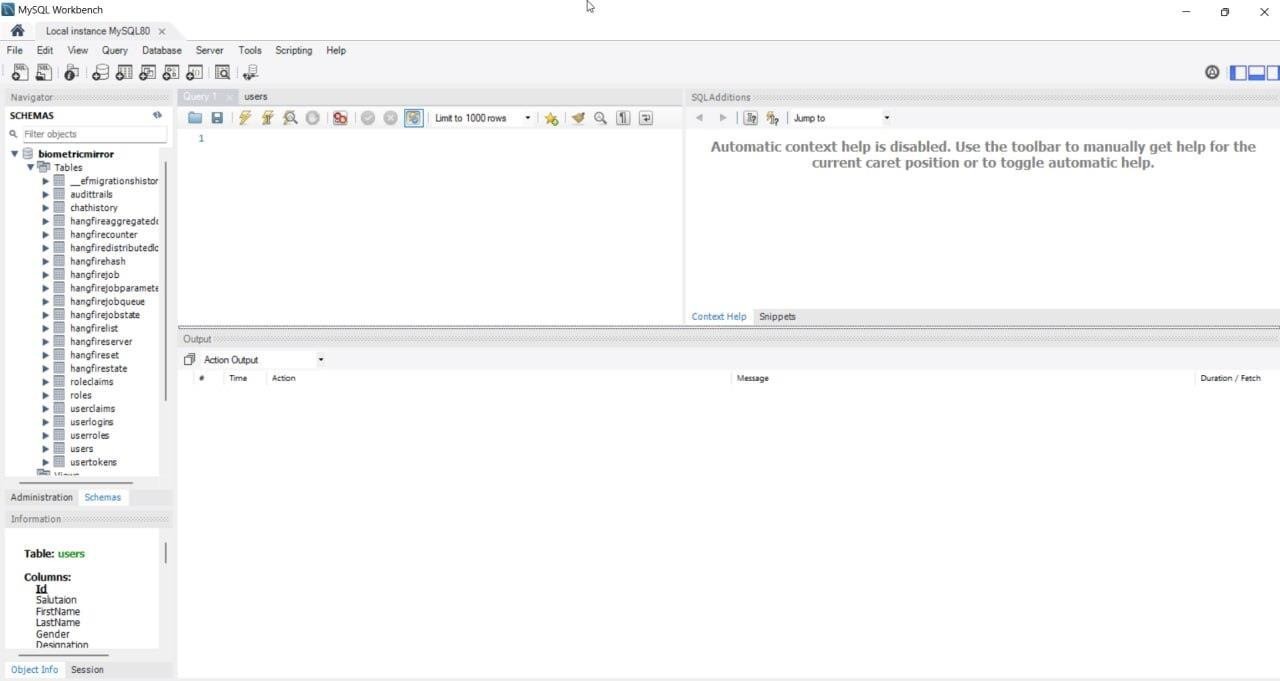
Password authentication is a common method used to verify the identity of a user in software applications or systems. It involves the user providing a password as a credential, which is then compared to a stored password associated with their account. If the provided password matches the stored password, the user is granted access.



**Figure 6.4 Password Authentication**

**STEP-4** DATABASE

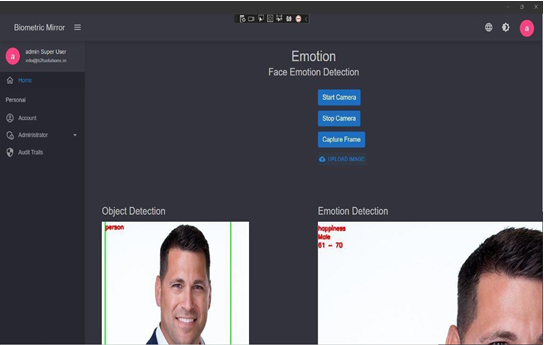
A database is a structured collection of data that is organized and managed to serve the needs of an application or system. Databases are used to store, retrieve, and manage large volumes of structured or unstructured data efficiently. They provide a structured way to organize and store data, ensuring data integrity, security, and scalability.



### Figure 6.5 Database

**STEP-5** FINAL OUTPUT

An age, gender, emotion , and object detection system's outputs It's crucial to remember that the age, gender, emotion, and object identification system's individual algorithms, models, and implementation specifics may change the final result.



**Figure 6.6 Final Output**

**CHAPTER 7**

**EXPERIMENTAL RESULTS**

* 1. **OVERVIEW**

To evaluate Biometric Mirror- Exploring Attitute Towards Facial and Object analysis. The predicted face emotions, age, gender and object Detection was measured using five metrics, namely, precision (Prec), recall (Rec), f1 score (F1), accuracy (Acc.). Let the letter be: Tp = True positive or occurrences where model predicted the positive truly, Tn = True negative or occurrences where model predicted the negative class truly, Fp = False positive or occurrences where model predicted the positive class falsely, Fn= False negative or occurrences where model predicted the negative class falsely, Precision, recall, accuracy, and f1 score shown in equations given below. The result and analysis are discussed.

## PERFORMANCE ANALYSIS

### Precision

Precision is defined as the ratio of true positive to the sum of true positive and false positive. It defines how often the classifier is correct when it predicts positive.

### Precision = TP / (TP + FP)

where,

TP = True Positives (Predicted as positive as was correct)

FP = False Positives (Predicted as positive but was incorrect)

### Recall

Recall is defined as the ratio of true positive to the sum of true positive and false negative [1]. It defines how the classifier is correct for all positive instances.

### Recall = TP / (TP + FP)

where,

TP = True Positives (Predicted as positive as was correct)

FN = False Negatives (Failed to predict an object that was there)

### F1 Score

The F1 score can be interpreted or defined as a weighted average of the precision and recall as given in the equation, where an F1 score has its best value at 1 and worst score at 0.

### F1 Score = 2\*(Precision\*Recall) / (Precision + Recall)

* + 1. **Accuracy**

To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

**Accuracy = TP+TN / TP+TN+FP+FN**

## RESULTS AND DISCUSSION

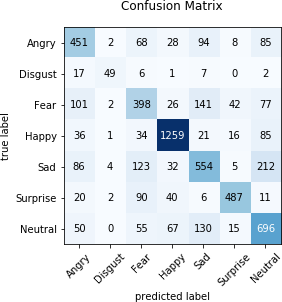
In this project, the proposed work of Biometric Mirror module is implemented using the DNN algorithm with the precision, Recall, F1-Score, Support.

## Performance analysis for Face emotions

### Table 7.1 Result Analysis of Face Emotions

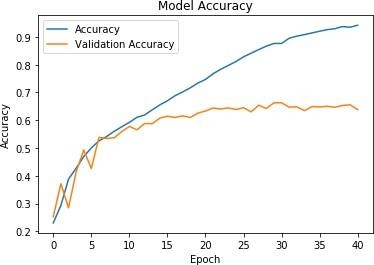
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Anger | Disgust | Fear | Happy | Sad | Suprise | Neutral |
| Precision | 0.59 | 0.81 | 0.51 | 0.86 | 0.64 | 0.84 | 0.59 |
| Recall | 0.61 | 0.59 | 0.50 | 0.86 | 0.54 | 0.74 | 0.70 |
| F1 score | 0.59 | 0.68 | 0.50 | 0.86 | 0.58 | 0.78 | 0.64 |
| Support | 736 | 60 | 774 | 1530 | 953 | 550 | 1170 |
| Accuracy | 70% | | | | | | |

The above table shows the proposed work of Face Emotions module which is implemented using the DNN algorithm. From the result, it shows that the DNN algorithm is very effective in classifying the appropriate emotions based on the FER-2013 Dataset.



### Figure 7.1 Confusion Matrix for seven emotions

In this model, the emotions are trained using DNN algorithm which provides a graphical representation of model accuracy.



### Figure 7.2 Confusion Matrix Model Accuracy

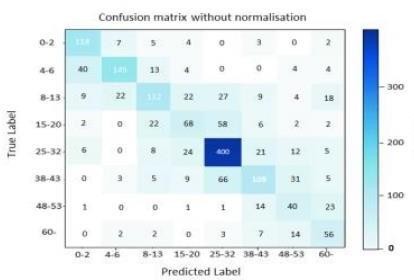
In this Face emotion detection 70 % percentage of the data set are used for training and 30% dataset are used for testing. This model is trained for 100 epochs with the batch size of 32 whose model accuracy is 70%

## Performance analysis for Age Prediction

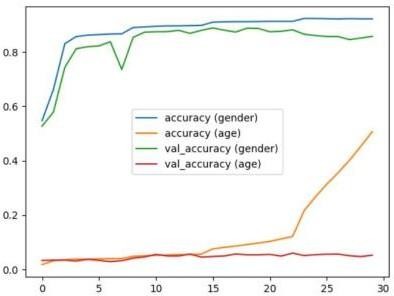
### Table 7.2 Result Analysis of Age Prediction

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F1 score** | **Support** | **Accuracy** |
| 0-2 | 0.70 | 0.86 | 0.77 | 159 | 65% |
| 4-6 | 0.81 | 0.69 | 0.74 | 210 |
| 8-13 | 0.67 | 0.50 | 0.57 | 223 |
| 15-20 | 0.39 | 0.42 | 0.40 | 2338 |
| 25-32 | 0.72 | 0.84 | 0.77 | 476 |
| 33-43 | 0.64 | 0.47 | 0.54 | 227 |
| 48-53 | 0.37 | 0.50 | 0.40 | 80 |
| 60-100 | 0.50 | 0.66 | 0.56 | 84 |

The above table shows the proposed work of Age Prediction module which is implemented using the DNN algorithm.

From the result, it shows that the DNN algorithm is very effective in classifying the appropriate based on the UTKFace Dataset.

### Figure 7.3 Confusion Matrix for Age Prediction



**Figure 7.4 Confusion Matrix Model Accuracy**

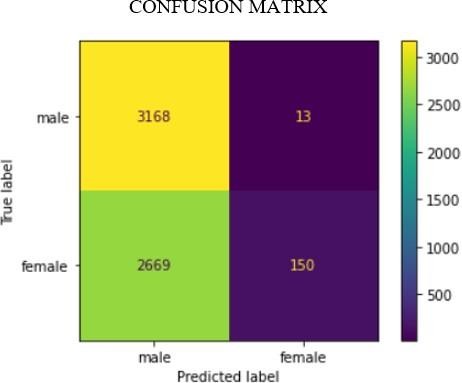
In this age detection 70 % percentage of the data set are used for training and 30% dataset are used for testing. This model is trained for 100 epochs with the batch size of 32 whose model accuracy is 65%

* + 1. **Performance analysis for Gender Prediction**

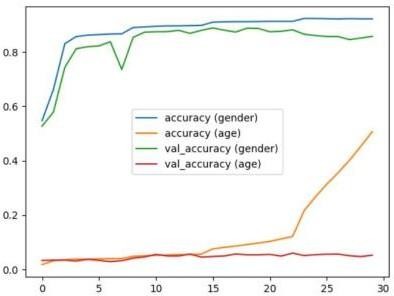
**Table 7.3 Result Analysis of Gender Prediction**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F1 score** | **Support** | **Accuracy** |
| **Male** | 0.54 | 0.99 | 0.70 | 3181 | 80% |
| **Female** | 0.92 | 0.05 | 0.10 | 2819 |

The above table shows the proposed work of Gender Prediction module which is implemented using the DNN algorithm. From the result, it shows that the DNN algorithm is very effective in classifying the appropriate gender based on the UTKFace Dataset.



**Figure 7.5 Confusion Matrix for Gender Prediction**



**Figure 7.6 Confusion Matrix Model Accuracy**

In this gender detection 70 % percentage of the data set are used for training and 30% dataset are used for testing. This model is trained for 100 epochs with the batch size of 32 whose model accuracy is 80%

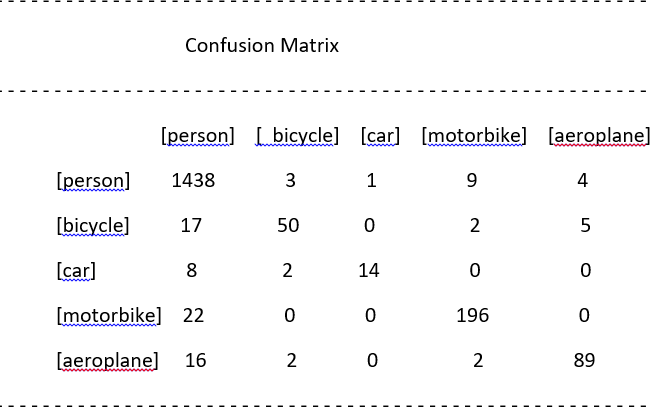
## Performance analysis for Object Detection

### Table 7.4 Result Analysis of Object Detection

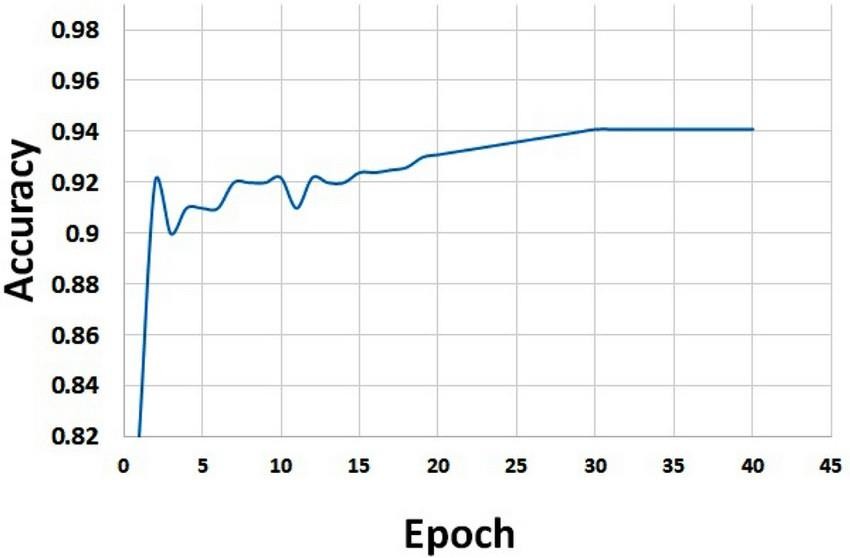
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Precision** | **Recall** | **F1 score** | **Support** | **Accuracy** |
| **Person** | 0.95 | 0.98 | 0.96 | 1455 | 94% |
| **Bicycle** | 0.87 | 0.67 | 0.75 | 74 |
| **Car** | 0.93 | 0.41 | 0.56 | 34 |
| **Motorbike** | 0.89 | 0.89 | 0.89 | 218 |
| **Aeroplane** | 0.90 | 0.81 | 0.85 | 109 |

The above table shows the proposed work of Object Detection module which is implemented using the YOLOv3 algorithm.

From the result, it shows that the YOLOv3 algorithm is very effective in classifying the appropriate objects based on the COCO Dataset.



### Figure 7.7 Confusion Matrix for Object Detection



**Figure 7.8 Confusion Matrix Model Accuracy**

In this Object detection 70 % percentage of the data set are used for training and 30% dataset are used for testing. This model is trained for 100 epochs with the batch size of 32 whose model accuracy is 94%

# **CHAPTER 8**

**CONCLUSION AND FUTURE ENHANCEMENTS**

In this project, we analyzed Bio-metric Mirror, an interactive facial and object analysis application that presented users with a personalized, speculative scenario of knowing themself psychologically. Deep Learning is an execution of Artificial Intelligence (AI). This study employs strengthen techniques. It enhance reputation price and execution time. The study involves Face Detection: Viola Jones Algorithm, Down Sampled: Fuzzy transform, Extracted characteristic: Ada Boost Technique, select characteristic: Stemmer Feature Wavelets decided on characteristic fed into DNN Classifier. It is network Trained by sample database FER-2013, UTKFace and COCO.

In this project, the proposed facial feature can be used to train a DNN for emotion recognition. Compared with other computer vision powered system, facial features can achieve similar accuracy as other machine learning algorithms (CNN). Yet, it reduces the data as well as the time required for training. The success of the face and object analysis project is evaluated using appropriate evaluation metrics, such as precision, recall, F1 score and accuracy. The accuracy of face emotions, age, gender and object detection is 70%, 65%, 80% and 94% respectively. Such advantages can significantly increase the speed of building applications involving emotion recognition. We found that users interpreted Biometric Mirror as a artifact that was capable of provoking reflection on the underlying concerns that are associated with facial analysis technology and automated decision-making.

Several future enhancements of facial and object detection are listed below:

* Ethical considerations
* Improved accuracy
* Real-time detection
* Multi-modal detection
* Robustness to variations

**REFERENCES**

1. J. Li and E. Y. Lam, "Facial expression recognition using deep neural networks," 2015 IEEE International Conference on Imaging Systems and Techniques (IST), Macau, China, 2015, pp. 1-6.
2. T. Khajontantichaikun, S. Jaiyen, S. Yamsaengsung, P. Mongkolnam and U. Ninrutsirikun, "Emotion Detection of Thai Elderly Facial Expressions using Hybrid Object Detection," 2022 26th International Computer Science and Engineering Conference (ICSEC), Sakon Nakhon, Thailand, 2022,pp.219-223, .https://ieeexplore.ieee.org/document/10049334
3. Yang, J. S. Y. Ortoneda and J. Saniie, "Emotion Recognition Using Deep Neural Network with Vectorized Facial Features," 2018 IEEE International Conference on Electro/Information Technology (EIT), Rochester, MI, USA, 2018, pp. 0318-0322.
4. Jaiswal, A. Krishnama Raju and S. Deb, "Facial Emotion Detection Using Deep Learning," 2020 International Conference for Emerging Technology (INCET), Belgaum, India, 2020, pp. 1-5
5. Lee, H. Jung, C. H. Ahn, J. Seo, J. Kim and O. Kwon, "Real-time personalized facial expression recognition system based on deep learning," 2016 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2016, pp. 267-268,.
6. Do, LN., Yang, HJ., Nguyen, HD. et al. Deep neural network-based fusion model for emotion recognition using visual data. J Supercomputer 77, 10773–10790 (2021).
7. W. Tarimo, M. M.Sabra and S. Hendre, "Real-Time Deep Learning-Based Object Detection Framework," 2020 IEEE Symposium Series on Computational Intelligence (SSCI)*,* Canberra, ACT, Australia, 2020, pp. 1829-1836
8. W. -F. Lin *et al*., "ONNC: A Compilation Framework Connecting ONNX to Proprietary Deep Learning Accelerators," *2019* IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*,* Hsinchu, Taiwan, 2019, pp. 214-218.
9. A. Mustafa and K. Meehan, "Gender Classification and Age Prediction using CNN and ResNet in Real-Time," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI)*,* Sakheer, Bahrain, 2020.

## APPENDIX -I

## Source Code

## BiometricMirror.API (SERVER PART)

<Project Sdk="Microsoft.NET.Sdk.Web">

<PropertyGroup>

<TargetFramework>net7.0</TargetFramework>

<Nullable>enable</Nullable>

<ImplicitUsings>enable</ImplicitUsings>

</PropertyGroup>

<ItemGroup>

<PackageReference Include="Hangfire.MySqlStorage" Version="2.0.3" />

<PackageReference Include="Swashbuckle.AspNetCore" Version="6.2.3" />

</ItemGroup>

<ItemGroup>

<Folder Include="Logs\" />

</ItemGroup>

<ItemGroup>

<PackageReference Include="FluentValidation.AspNetCore" Version="11.2.1" />

<PackageReference Include="Hangfire" Version="1.7.31" />

<PackageReference Include="LazyCache.AspNetCore" Version="2.4.0" />

<PackageReference Include="Microsoft.AspNetCore.Authentication.JwtBearer" Version="6.0.10" />

<PackageReference Include="Microsoft.AspNetCore.Mvc.Versioning" Version="5.0.0" />

<PackageReference Include="Microsoft.AspNetCore.Mvc.Versioning.ApiExplorer" Version="5.0.0" />

<PackageReference Include="Microsoft.EntityFrameworkCore.Tools" Version="6.0.10">

<PrivateAssets>all</PrivateAssets>

<IncludeAssets>runtime; build; native; contentfiles; analyzers; buildtransitive</IncludeAssets>

</PackageReference>

<PackageReference Include="Swashbuckle.AspNetCore" Version="6.4.0" />

<PackageReference Include="Serilog.Enrichers.Environment" Version="2.2.0" />

<PackageReference Include="Serilog.Extensions.Hosting" Version="5.0.1" />

<PackageReference Include="Serilog.Formatting.Compact" Version="1.1.0"/>

<PackageReference Include="Serilog.Settings.Configuration" Version="3.4.0"/>

<PackageReference Include="Serilog.Sinks.Async" Version="1.5.0" />

<PackageReference Include="Serilog.Sinks.Console" Version="4.1.0" />

<PackageReference Include="Serilog.Sinks.File" Version="5.0.0" />

<PackageReference Include="Emgu.CV" Version="4.6.0.5131" /

<PackageReference Include="Emgu.CV.Bitmap" Version="4.6.0.5131" />

<PackageReference Include="Emgu.CV.runtime.windows" Version="4.6.0.5131" />

<PackageReference Include="Microsoft.ML" Version="2.0.1" />

<PackageReference Include="Microsoft.ML.OnnxRuntime" Version="1.14.1"/>

<PackageReference Include="Microsoft.ML.OnnxTransformer"

Version="2.0.1" />

<PackageReference Include="Microsoft.ML.Vision" Version="2.0.0" />

<PackageReference Include="SciSharp.TensorFlow.Redist" Version="2.3.1"/>

</ItemGroup>

<ItemGroup>

<ProjectReference Include="..\BiometricMirror.Application\BiometricMirror.Application.csproj" />

<ProjectReference Include="..\BiometricMirror.Infrastructure\BiometricMirror.Infrastructure.csproj" />

</ItemGroup>

<ItemGroup>

<None Update="haarcascade\_frontalface\_default.xml">

<CopyToOutputDirectory>Always</CopyToOutputDirectory>

</None>

</ItemGroup>

</Project>

**BiometricMirror.MAUI (CLIENT PART)**

<Project Sdk="Microsoft.NET.Sdk.Razor">

<PropertyGroup>

<TargetFrameworks>net7.0-android;net7.0-ios;net7.0- maccatalyst</TargetFrameworks>

<TargetFrameworks Condition="$([MSBuild]::IsOSPlatform('windows'))">$(TargetFrameworks);net7.0- windows10.0.19041.0</TargetFrameworks>

<!-- Uncomment to also build the tizen app. You will need to install tizen by following this: https://github.com/Samsung/Tizen.NET -->

<!-- <TargetFrameworks>$(TargetFrameworks);net6.0-tizen</TargetFrameworks> -->

<OutputType>Exe</OutputType>

<RootNamespace>BiometricMirror.MAUI</RootNamespace>

<UseMaui>true</UseMaui>

<SingleProject>true</SingleProject>

<ImplicitUsings>enable</ImplicitUsings>

<EnableDefaultCssItems>false</EnableDefaultCssItems>

<!-- Display name -->

<ApplicationTitle>BiometricMirror.MAUI</ApplicationTitle>

<!-- App Identifier -->

<ApplicationId>com.companyname.BiometricMirror.maui</ApplicationId>

<ApplicationIdGuid>DC83BDAB-C168-4079-86EC- 35D27253B5AC</ApplicationIdGuid>

<!-- Versions -->

<ApplicationDisplayVersion>1.0</ApplicationDisplayVersion>

<ApplicationVersion>1</ApplicationVersion>

<SupportedOSPlatformVersion Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'ios'">14.2</SupportedOSPlatformVersion>

<SupportedOSPlatformVersion Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'maccatalyst'">14.0</SupportedOSPlatformVersion>

<SupportedOSPlatformVersion Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'android'">24.0</SupportedOSPlatformVersion>

<SupportedOSPlatformVersion Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'windows'">10.0.17763.0</SupportedOSPlatformVersion>

<TargetPlatformMinVersion Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'windows'">10.0.17763.0</TargetPlatformMinVersion>

<SupportedOSPlatformVersion Condition="$([MSBuild]::GetTargetPlatformIdentifier('$(TargetFramework)')) == 'tizen'">6.5</SupportedOSPlatformVersion>

</PropertyGroup>

<ItemGroup>

<!-- App Icon -->

<MauiIcon Include="Resources\AppIcon\appicon.svg" ForegroundFile="Resources\AppIcon\appiconfg.svg" Color="#512BD4" />

<!-- Splash Screen -->

<MauiSplashScreen Include="Resources\Splash\splash.svg" Color="#512BD4" BaseSize="128,128" />

<!-- Images -->

<MauiImage Include="Resources\Images\\*" />

<MauiImage Update="Resources\Images\dotnet\_bot.svg" BaseSize="168,208" />

<!-- Custom Fonts -->

<MauiFont Include="Resources\Fonts\\*" />

<!-- Raw Assets (also remove the "Resources\Raw" prefix) -->

<MauiAsset Include="Resources\Raw\\*\*" LogicalName="%(RecursiveDir)%(Filename)%(Extension)" />

</ItemGroup>

<ItemGroup>

<None Include="Shared\MainLayout.razor" />

<None Include="Shared\NavMenu.razor" />

<None Include="Shared\SurveyPrompt.razor" />

<None Include="wwwroot\css\bootstrap\bootstrap.min.css.map" />

<None Include="wwwroot\css\open-iconic\FONT-LICENSE" />

<None Include="wwwroot\css\open-iconic\font\fonts\open-iconic.svg" />

<None Include="wwwroot\css\open-iconic\ICON-LICENSE" />

<None Include="wwwroot\css\open-iconic\README.md" />

</ItemGroup>

<ItemGroup>

<ProjectReference Include="..\BiometricMirror.Application\BiometricMirror.Application.csproj" />

<ProjectReference Include="..\BiometricMirror.Infrastructure\BiometricMirror.Infrastructure.csproj" />

<ProjectReference Include="..\BiometricMirror.UI.Infrastructure\BiometricMirror.UI.Infrastructure.csproj" />

</ItemGroup>

<ItemGroup>

<PackageReference Include="Blazored.FluentValidation" Version="2.0.3" />

<PackageReference Include="Microsoft.AspNetCore.Components.Authorization" Version="6.0.10" />

<PackageReference Include="Microsoft.AspNetCore.Identity" Version="2.2.0" />

<PackageReference Include="Microsoft.AspNetCore.SignalR.Client" Version="6.0.10" />

<PackageReference Include="Microsoft.Extensions.Http" Version="6.0.0" />

<PackageReference Include="Microsoft.Extensions.Http.Polly" Version="6.0.10" />

<PackageReference Include="Microsoft.Extensions.Identity.Core" Version="6.0.10" />

<PackageReference Include="Microsoft.Extensions.Localization" Version="6.0.10" />

<PackageReference Include="MudBlazor" Version="6.0.17" />

<PackageReference Include="Plugin.Fingerprint" Version="3.0.0-beta.1" />

<PackageReference Include="SixLabors.Fonts" Version="1.0.0-beta15" />

<PackageReference Include="SixLabors.ImageSharp" Version="1.0.4" />

<PackageReference Include="SixLabors.ImageSharp.Drawing" Version="1.0.0-beta13" />

<PackageReference Include="System.Net.Http.Json" Version="6.0.0" />

<PackageReference Include="Toolbelt.Blazor.HttpClientInterceptor" Version="10.2.0" />

</ItemGroup>

<ItemGroup>

<Compile Update="Resources\MlModels\ModelResourceresx.Designer.cs">

<DesignTime>True</DesignTime>

<AutoGen>True</AutoGen>

<DependentUpon>ModelResourceresx.resx</DependentUpon>

</Compile>

</ItemGroup>

<ItemGroup>

<EmbeddedResource Update="Resources\MlModels\ModelResourceresx.resx">

<Generator>ResXFileCodeGenerator</Generator>

<LastGenOutput>ModelResourceresx.Designer.cs</LastGenOutput>

</EmbeddedResource>

</ItemGroup>

</Project>

**BiometricInfrastructure.cs**

using BiometricMirror.Infrastructure.Contexts; using BiometricMirror.Infrastructure.Helpers;

using BiometricMirror.Infrastructure.Models.Identity; using Microsoft.AspNetCore.Identity;

using Microsoft.Extensions.Localization; using Microsoft.Extensions.Logging;

using BiometricMirror.Application.Interfaces.Services; using BiometricMirror.Shared.Constants.Permission; using BiometricMirror.Shared.Constants.Role;

using BiometricMirror.Shared.Constants.User;

namespace BiometricMirror.Infrastructure

{

public class DatabaseSeeder : IDatabaseSeeder

{

private readonly ILogger<DatabaseSeeder> \_logger;

private readonly IStringLocalizer<DatabaseSeeder> \_localizer; private readonly ApplicationDbContext \_db;

private readonly UserManager<ApplicationUser> \_userManager; private readonly RoleManager<ApplicationRole> \_roleManager;

public DatabaseSeeder( UserManager<ApplicationUser> userManager, RoleManager<ApplicationRole> roleManager, ApplicationDbContext db, ILogger<DatabaseSeeder> logger, IStringLocalizer<DatabaseSeeder> localizer)

{

\_userManager = userManager;

\_roleManager = roleManager;

\_db = db;

\_logger = logger;

\_localizer = localizer;

}

public void Initialize()

{

AddAdministrator();

//AddBasicUser();

\_ = \_db.SaveChanges();

}

private void AddAdministrator()

{

Task.Run(async () =>

{

//Check if Role Exists

ApplicationRole adminRole = new(RoleConstants.AdministratorRole,

\_localizer["Administrator role with full permissions"]); ApplicationRole? adminRoleInDb = await

\_roleManager.FindByNameAsync(RoleConstants.AdministratorRole); if (adminRoleInDb == null)

{

\_ = await \_roleManager.CreateAsync(adminRole); adminRoleInDb = await

\_roleManager.FindByNameAsync(RoleConstants.AdministratorRole);

\_logger.LogInformation(\_localizer["Seeded Administrator Role."]);

}

//Check if User Exists ApplicationUser superUser = new()

{

FirstName = "Super", LastName = "User",

Email = "[info@b2lsolutions.in",](mailto:info@b2lsolutions.in) UserName = "admin", EmailConfirmed = true, PhoneNumberConfirmed = true, CreatedOn = DateTime.Now, IsActive = true

};

ApplicationUser superUserInDb = await

\_userManager.FindByEmailAsync(superUser.Email); if (superUserInDb == null)

{

\_ = await \_userManager.CreateAsync(superUser, UserConstants.DefaultPassword);

IdentityResult result = await \_userManager.AddToRoleAsync(superUser, RoleConstants.AdministratorRole);

if (result.Succeeded)

{

\_logger.LogInformation(\_localizer["Seeded Default SuperAdmin User."]);

}

else

{

foreach (IdentityError? error in result.Errors)

{

}

**DnnPredictions.cs**

using Emgu.CV.CvEnum; using Emgu.CV;

using Emgu.CV.Dnn; using Emgu.CV.Structure; using Emgu.CV.Util; using System;

using System.Collections;

using System.Collections.Generic; using System.Linq;

using System.Text;

using System.Threading.Tasks; using Emgu.CV.DepthAI; using System.Diagnostics; using Emgu.CV.Flann;

using Emgu.CV.Face;

using System.Drawing.Imaging; using System.Drawing;

using BiometricMirror.Application.Responses; using BiometricMirror\_API;

namespace BiometricMirror.API

{

public class DnnPrediction

{

private CascadeClassifier haarCascade; private int LineThickness = 2;

private int Padding = 10;

public List<string> Labels = new List<string>() { "neutral", "happiness", "surprise", "sadness", "anger", "disgust", "fear", "contempt" };

private Image<Gray, Byte> detectedFace = null;

private readonly List<string> \_genderList = new List<string> { "Male", "Female" }; private readonly List<string> \_ageList = new List<string> { "(0-2)", "(4-6)", "(8-12)",

"(15-20)", "(25-32)", "(38-43)", "(48-53)", "(60-100)" };

public DnnPrediction()

{

try

{

haarCascade = new CascadeClassifier("haarcascade\_frontalface\_default.xml");

}

catch (Exception ex)

{

}

Image = Convert.ToBase64String(GetBytesFromBitmap(imageOutput.ToBitmap()))

};

}

private byte[] GetBytesFromBitmap(Bitmap bitmap)

{

using (var m = new MemoryStream())

{

bitmap.Save(m, ImageFormat.Bmp); return m.ToArray();

}

}

private void GetMaxClass(Mat probBlob, out int classId)

{

// reshape the blob to 1×1000 matrix

using var probMat = probBlob.Reshape(1, 1); double minval = 0, maxval = 0;

Point min = new Point(), Max = new Point();

CvInvoke.MinMaxLoc(probMat, ref minval, ref maxval, ref min, ref Max); classId = min.X;

// Debug.WriteLine($"X: {classNumber.X} – Y: {classNumber.Y} ");

}

private List<float[]> Array2DList(Array array)

{

IEnumerator enumerator = array.GetEnumerator(); var row = array.GetLength(0);

var col = array.GetLength(1); var result = new List<float[]>(); var temp = new List<float>(); for (int i = 0; i < row; i++)

{

temp.Clear();

for (int j = 0; j < col; j++)

{

temp.Add(float.Parse(array.GetValue(i, j).ToString()));

}

result.Add(temp.ToArray());

}

return result;

}

}

}

**APPENDIX -II**

**List of Publications**