BINDURA UNIVERSITY OF SCIENCE EDUCATION

FACULTY OF SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE

Offline Facial Recognition Using Machine Learning Techniques For Exammination Attendance & Logging

By

Ernest Taonga Bhebe

REG NUMBER : B1645264

SUPERVISOR : Mr. T. Mhlanganiso

A RESEARCH PROJECT SUBMITTED TO THE COMPUTER SCIENCE DEPARTMENT IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE BACHELOR OF SCIENCE HONOURS DEGREE IN COMPUTER SCIENCE.

29 NOVEMER 2023

APPROVAL FORM

The undersigned certify that they have supervised the student Ernest Taonga Bhebe's dissertation entitled, "**Offline facial recognition using machine learning techniques for examination attendance and logging**" is submitted in partial fulfillment of the requirements for a Bachelor of Computer Science Honors Degree at Bindura University of Science Education.

STUDENT: **DATE:**

30/11/23

………………………………………………… …………………………...

SUPERVISOR: **DATE:**

30/11/23

………………………………………………… ..…………………………..

CHAIRPERSON: **DATE:**

30/11/23

……………………………………….………… ..…………………………..

EXTERNAL EXAMINER: **DATE:**

………………………………………..……………   ..……………………………

Abstract

This research project aimed to develop a machine learning-powered Android application for offline facial authentication of students during exams, attendance marking, and break logging. The mobile application is designed to capture and process facial features for subsequent recognition using a deep learning model. The proposed system comprises several processing stages, including image processing, feature extraction, and facial recognition. The facial recognition model integrated into the mobile application is based on a convolutional neural network (CNN) architecture trained on a large dataset of image samples.

Dedication

This research is dedicated to my family and relatives, who have been a constant source of nurturing and encouragement from the very beginning. I am aware that they have eagerly awaited the day when I would achieve my academic goals. I take great pride in being their son. I am eternally grateful for their unwavering support and unwavering motivation throughout my academic journey. Finally, I extend my heartfelt gratitude to my classmates for their unconditional love and unwavering support. I am deeply touched by the belief you have placed in my dreams.

ACKNOWLEDGEMENTS

I thank the Lord Almighty for bring me this far. I would like to thank all the student's and lecturers at Bindura University of Science Education, especially Mr. T. Mhlanganiso (Project supervisor), Mr. D. Ndumiyana (Industrial attachment supervisor), Mr P. Chaka (Chairperson) and the rest of the Lecturers in the Faculty of Science and Engineering. I am deeply grateful for the dedication and commitment demonstrated by all those who contributed to the success of my research. I would like to extend my sincere appreciation to Bindura University of Science Education for providing the necessary infrastructure and academic resources that were instrumental in the completion of this study. Additionally, I am immensely thankful for the unwavering support and encouragement provided by my parents and family members throughout this endeavor. While there are others whose names, I may not be able to recall at this moment, I am equally grateful for their contributions and assistance. Your collective support has been invaluable, and I am truly humbled by your generosity.

Table of Contents

Chapter 1: Problem Identification

This chapter will be concentrating on the issues and problems that have contributed to the development of this project.

1.1 Introduction

Recent advancements in machine learning have opened up new possibilities in various fields, particularly in computer vision. Among the rapidly evolving areas within computer vision is facial recognition, which holds immense potential for applications in various sectors, including education and banking. However, real-time facial recognition from images presents a challenging task due to inherent variations in facial features, hairstyles, noise, and diverse image acquisition conditions. This research project delves into the offline facial recognition of human physical facial features on images using deep learning techniques. Offline recognition refers to the ability to recognize features without relying on real-time communication with an online server. By harnessing the power of deep learning algorithms, which have demonstrated remarkable capabilities in image processing and character recognition, this project aims to develop a robust and efficient mobile application for facial authentication from images.

The proposed approach will utilize a deep learning architecture known as a convolutional neural network (CNN), which has demonstrated exceptional effectiveness in handling image data. CNNs possess the remarkable ability to automatically learn and extract relevant features from images, enabling them to recognize intricate patterns and structures. By training the CNN on a vast dataset of portrait images, the model will acquire the ability to gene-ralize and accurately recognize faces. This project strives to achieve a high degree of accuracy in facial recognition while prioritizing computational efficiency to ensure real-time or near real-time performance.

1.2 Background of study

In the decade following 2013, mobile devices experienced a remarkable transformation, becoming more powerful, secure, intelligent, efficient, and compact in size. This evolution led to an increased reliance on these devices for a wide range of tasks, including data capture, sharing, and leveraging their advanced computing power. Recognizing this trend, the proposed project aims to harness the capabilities of mobile devices within educational institutions, enabling invigilators to effectively identify and verify students, log exam events, and facilitate long-term data storage.

Bindura University of Science Education has a student population of approximately 5,000 to 5,999 students, as reported on *www.4icu.org/reviews/9561.htm*. This large student body poses a challenge during examination periods, particularly when it comes to verifying student identities, checking attendance, and recording events such as breaks. To address this challenge and streamline the examination process, there is an urgent need for an efficient and reliable method for identifying and recording student-related events during exams. This method should accurately verify student identities using biometric authentication, store large amounts of data quickly and accurately, and provide long-term data storage while utilizing a compact storage device that is smaller than a wallet.

The current methods of data capturing and student identity verification are inadequate and inefficient. Identity verification relies on institute-issued ID cards, while data capture involves the use of pens and paper. This approach leads to several drawbacks. The moderate security level of ID cards allows some students to exploit the system and impersonate others. Data capture using pens and paper consumes large amounts of materials, requiring extensive storage space. Additionally, this method is susceptible to human error. Paper-based data storage requires extensive physical space and is cumbersome for analyzing student behavior patterns. Additionally, it prolongs the time spent by invigilators and students on tasks such as marking attendance, logging student activities, and verifying identities.

1.3 Statement of The Problem

The current methods for collecting data and verifying student identities are not efficient and effective in handling a large number of students. During exams, the process of checking IDs is time-consuming and prone to errors due to the reliance on school-issued ID cards, which can be easily forged or misused. Additionally, taking attendance is labor-intensive as each student must manually record their information on paper forms, which then need to be collected and processed. Furthermore, the paper-based system makes it challenging to extract valuable insights from the gathered data, hindering data analysis and decision-making.

1.4 Research Objectives

- To analyze different machine learning techniques used for facial recognition.
- Conduct a survey on students and invigilators to identify application requirements.
- To design and implement an android application for recognizing facial features.

1.5 Research Questions

- What are the different machine learning techniques used for facial recognition?
- Evaluate the effectiveness of machine learning techniques in recognizing facial features on images?
- Which is the best and feasible way to test the application and success of the research?

1.6 Research Propositions/Hypothesis

- H_0: The android application will be able to detect and recognize the facial features.
- H_2: The android application will be able to store data long term and provide output using easy to understand graphical representation methods.
- H_3: The android application will prioritize computational efficiency to ensure real-time or near real-time performance.

1.7 Justification/Significance of The Study

The proposed research project holds great promise for enhancing the accuracy and reliability of student identity verification during examinations, with far-reaching applications across various academic institutions and resource-scarce settings. Leveraging machine learning techniques, the project seeks to develop an efficient and precise offline facial recognition mobile application for widespread adoption in educational settings. This groundbreaking endeavor will revolutionize current security measures, modernize data storage practices, and refine data analytical techniques. The project's findings will significantly expand our understanding of facial recognition as a biometric authentication method.

1.8 Assumptions

- Every lecturer owns and uses and android based device with OS version 5.0 to newer.
- Every android device has free storage of up 240MB.
- Both front and back cameras for the devices should at least have 8-12MP.
- Device equipped with a processor chip-set with 4 or more cores.
- Device RAM should at least be 2GB.
- Built in storage on device should at least be 32GB.

1.9 Limitations/challenges

- **Resource constraints** - Due to the proposed system's reliance on offline operation, its performance may be influenced by the hardware and computational resources available.
- **Timeframe constraints** - The proposed research project dissertation will be completed within a tight timeframe. This limitation might impact the scope and depth of the study conducted, as well as the effectiveness of the proposed system.
- **Image quality constraint** - Low image resolution and noise can pose significant challenges for reliable facial feature recognition.
- **Geographic constraint** - The proposed research project will focus primarily on Bindura University of Science Education, located in Bindura, Zimbabwe.

1.10 Scope/Delimitation of the Research

The overarching objective of this research project is to develop a robust and accurate machine learning-powered offline facial recognition application. The application will be designed to operate independently of an active internet connection, making it suitable for implementation in resource-constrained environments. The primary focus of the project will be on its application at Bindura University of Science Education, located in Bindura town. During examinations, invigilators will utilize the mobile application to verify student identities, mark attendance, record latecomers, flag disqualified students, track exam duration, and document toilet or water breaks.

1.11 Definition of Terms

- ✓ **Machine learning (ML):** is a field of artificial intelligence (AI) that focuses on the development and study of algorithms that can learn from data without being explicitly programmed.
- ✓ **Convolutional Neural Networks (CNNs):** are a type of artificial neural network (ANN) that is particularly well-suited for processing image data. They are able to extract and learn high-level features from images, such as edges, textures, and shapes.
- ✓ **Deep learning:** is a subset of machine learning that uses artificial neural networks (ANNs) with multiple layers. ANNs are inspired by the structure of the human brain, with layers of interconnected nodes that process and transmit information.
- ✓ **Megabyte (MB):** is a measurement of storage in relation to bytes where 1MB is 1048576 bytes and **Gigabyte (GB):** is also a measurement of storage in relation to bytes where 1GB is 1073741824 bytes.
- ✓ **Megapixels (MP):** are units of measurement that indicates how many pixels are in an image where pixels are small photo sensitive elements that are arranged in grids of rows and columns. They determine the quality of the image.
- ✓ **User Interface (UI):** the visual part of a software that a user sees and interacts with.
- ✓ Chip-set/Processor: processor, also known as a central processing unit (CPU), is the main computational unit of a computer system. It is responsible for executing instructions and performing calculations.

- ✓ **Operating System (OS):** a software that provide users with an interface to use and electronic devices and at the same time controls allocation of hardware resources such as memory, processing time etc.
- ✓ **Random Access Memory (RAM):** a volatile computer memory that stores program and data during processing.
- ✓ **Processor cores:** are separate processing units within a single integrated circuit. Each core is capable of executing instructions independently, these units can be plugged together into one single integrated circuit to work as one multi-cored processors.
- ✓ **Dataset:** is a collection of data that is organized and stored for analysis. Datasets can be used to train machine learning models, analyze trends, and make predictions. They can be structured or unstructured, and they can be collected from a variety of sources, such as surveys, experiments, and social media.
- ✓ **Database:** is a structured collection of data that is organized and stored electronically. Databases are designed to store, retrieve, and manage data efficiently.

Chapter 2: Literature Review

According to the University of Edinburgh at https://www.ed.ac.uk/institute-academic-development/study-hub/learning-resources/literature-review, A literature review is a piece of academic writing demonstrating knowledge and understanding of the academic literature on a specific topic placed in context. A literature review also includes a critical evaluation of the material; this is why it is called a literature review rather than a literature report. It is a process of reviewing the literature, as well as a form of writing. To illustrate the difference between reporting and reviewing, think about television or film review articles. These articles include content such as a brief synopsis or the key points of the film or program plus the critic's own evaluation. Similarly, the two main objectives of a literature review are firstly the content covering existing research, theories and evidence, and secondly your own critical evaluation and discussion of this content.

2.1 Introduction

Facial recognition has been a topic of research for several decades, with significant progress made in recent years using machine learning techniques. Early methods for recognizing facial features involved human experts manually identifying and labeling facial features, such as the eyes, nose, and mouth. This was a time-consuming and error-prone process, but it was the only method available until the development of more sophisticated automated techniques in the late 20th century. However, with the advent of machine learning techniques, significant progress has been made in the field computer vision. This chapter aims to provide an overview of the current state-of-the-art techniques for recognizing faces using machine learning algorithms, with a focus on offline recognition. The review will explore the different machine learning algorithms and deep learning models that have been used for facial authentication and their effectiveness in offline recognition. It will also examine the different preprocessing, feature extraction, and classification techniques that have been used in the literature to develop accurate and efficient recognition systems. The literature review will cover a diverse range of studies, including recent research papers, books, and reviews. It will also compare the different approaches used in the literature and identify the strengths and weaknesses of each approach.

2.2 Facial Features

Facial recognition is a biometric technology that uses facial features to identify or verify an individual. It is one of the most popular biometric technologies because it is non-invasive and can be used from a distance. Facial recognition systems work by extracting and analysing facial features from an image or video. The extracted features are then compared to a database of known faces to identify or verify an individual.

Some of the key facial features used for facial recognition include:

- Distance between the eyes: The distance between the eyes is one of the most important facial features for facial recognition. It is a relatively stable feature that is not easily affected by changes in facial expression or lighting.
- Distance between the nose and mouth: The distance between the nose and mouth is another important facial feature for facial recognition. It is also a relatively stable feature that is not easily affected by changes in facial expression or lighting.
- Shape of the nose: The shape of the nose is a unique facial feature that can be used to identify an individual. The nose is made up of several different bones and cartilage, which gives it a complex shape.
- Shape of the ears: The shape of the ears is another unique facial feature that can be used to identify an individual. The ears are made up of several different pieces of cartilage, which gives them a complex shape.
- Shape of the lips: The shape of the lips is a unique facial feature that can be used to identify an individual. The lips are made up of several different muscles, which gives them a complex shape.

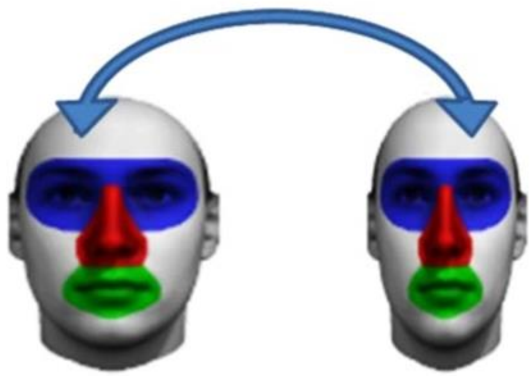


Figure 1: Facial features

In addition to these key facial features, facial recognition systems may also use other features such as the shape of the cheekbones, the depth of the eye sockets, and the contour of the chin. The more facial features that are used, the more accurate the facial recognition system will be.

2.3 Facial Authentication

Facial authentication is a biometric security method that uses a person's facial features to verify their identity. This technology is becoming increasingly popular in a variety of applications, including: Smartphones and tablets: Facial authentication is being used to unlock smartphones and tablets, as well as to authorize payments and access to sensitive applications. Physical access control: Facial authentication is being used to control access to buildings, secure areas, and events. Border control: Facial authentication is being used to verify the identity of travellers at airports and other border crossings. Law enforcement: Facial authentication is being used to identify suspects and criminals. Facial authentication is a convenient and secure way to verify identity.

Advantages:

- Convenience: Facial authentication is a quick and easy way to verify identity.
- Security: Facial authentication is a secure way to verify identity, as it is difficult to forge or replicate a person's face.

- Accuracy: Facial authentication systems are becoming increasingly accurate, with some systems achieving accuracy rates of over 99%.

Disadvantages:

- Privacy: Facial authentication raises privacy concerns, as it involves the collection and storage of sensitive facial data.
- Bias: Facial authentication systems can be biased against certain groups of people, such as people of colour, women, and children.
- Error: Facial authentication systems can be fooled by certain techniques, such as using makeup or masks.

Overall, facial authentication is a promising technology with a wide range of applications. However, it is important to address the privacy and bias concerns associated with this technology before it is widely adopted.

2.4 Machine Learning

Machine learning, a branch of AI, specializes in developing algorithms and models that allow computers to learn and make predictions or decisions without being explicitly programmed. It seeks to create systems that can continuously learn and improve from experience or data. Machine learning revolves around constructing mathematical models that can extract patterns and connections from data. These models are trained using a dataset comprising input variables (features) and corresponding output variables (labels or targets). By scrutinizing the data, machine learning algorithms can pinpoint patterns, make predictions, or categorize new data. There are several categories of machine learning algorithms, namely supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves training a model on labelled examples, where the input data is accompanied by the associated correct output labels. The algorithm learns to map the input to the output by uncovering patterns and correlations in the training data. Supervised learning is a type of machine learning that uses labelled data to train algorithms to make predictions. It is commonly used for tasks such as classification and regression. Classification involves predicting a discrete category, such as whether an image contains a cat or a dog. Regression involves predicting a continuous value, such as the price of a house.

Unlike supervised learning, unsupervised learning algorithms operate on unlabelled data, where only the input variables are provided. The aim is to uncover hidden patterns or structures within the data without the need for predetermined labels. Clustering, dimensionality reduction, and anomaly detection are examples of unsupervised learning tasks. Reinforcement learning has been successfully applied to a wide variety of problems, including game playing, robotics, and self-driving cars. For example, the DeepMind team used reinforcement learning to create AlphaGo, which became the first computer program to defeat a world champion at the game of Go. Machine learning algorithms can be subdivided into distinct subcategories, including decision trees, random forests, support vector machines (SVMs), neural networks, and others. Each algorithm possesses unique strengths and limitations, and the selection of an algorithm hinges on the specific problem being addressed, the available data, and the desired outcome. Machine learning has found applications across a multitude of domains, including healthcare, finance, natural language processing, computer vision, recommendation systems, and many more. Its ability to autonomously learn from data and generate predictions or conclusions has rendered it a potent tool for tackling complex problems and extracting valuable insights from extensive datasets.

2.5 Deep Learning

Deep learning, a subset of machine learning, centres on training artificial neural networks with multiple layers, also known as deep neural networks, to enable them to learn and make predictions or decisions. Deep learning algorithms draw inspiration from the structure and function of the human brain, particularly the interconnected network of neurons. Deep neural networks, also known as deep nets, are intricate structures composed of multiple layers of interconnected nodes, termed artificial neurons or units. Each unit receives input data, applies a mathematical transformation or activation function, and transmits the resulting output to the subsequent layer. The depth of the network is determined by the number of hidden layers sandwiched between the input and output layers. Deep learning models can encompass tens, hundreds, or even thousands of layers. Deep learning's distinguishing characteristic is its capacity to autonomously acquire hierarchical representations of data. Each layer within the deep neural network proficiently extracts higher-level features or representations from the input data. Lower layers focus on extracting fundamental features like edges or textures, while higher layers specialize in extracting more intricate features and patterns. By fusing together these acquired features, deep learning models can grasp intricate relationships and make highly accurate predictions.

Convolutional Neural Networks (CNNs) stand among the widely utilized deep learning architectures. CNNs are predominantly employed for image and video analysis. They leverage specialized layers, such as convolutional layers, pooling layers, and fully connected layers, to automatically learn and extract spatial hierarchies of features from images. Recurrent Neural Networks (RNNs) are specialized neural networks tailored to process sequential data, including time series data and natural language processing tasks. They incorporate feedback connections that facilitate the flow of information in cycles, enabling the network to retain and utilize information from preceding steps or time points. Generative Adversarial Networks (GANs) represent a unique type of neural network architecture that involves two competing networks: a generator and a discriminator. These networks are trained together in a collaborative yet adversarial manner to achieve remarkable results.

Deep learning's transformative impact has revolutionized various fields, including computer vision, speech recognition, natural language processing, and recommendation systems. Its profound advancements have propelled the state-of-the-art in tasks such as image classification, object detection, image generation, machine translation, and a plethora of others. Deep learning models, despite their remarkable capabilities, often demand copious amounts of labelled training data and substantial computational resources to achieve optimal performance. They are also susceptible to over-fitting, a phenomenon where the model memorizes the training data and struggles to generalize effectively to new, unseen data. Deep learning's transformative power has revolutionized numerous fields and continues to drive advancements in artificial intelligence by enabling more accurate and intricate modelling of data. To overcome the challenges inherent in deep learning, regularization techniques and data augmentation are frequently employed.

2.6 Related Literature

(Mutiu, et al, 2012) proposed a Biometric Model for Examination Screening and Attendance Monitoring in Yaba College of Technology which used fingerprint to Screen and Attendance Monitoring using Finite State Automata Theory. The students would only need to touch the sensor so that their fingerprint can be validated against a database thus showing their true identity. The advantages include Efficiency in Student Screening Unlike the manual method of screening a student does not have to go through the rigorous process of identification before they are allowed into the examination Hall. Their identity information is already stored in the database which they have to confirm ownership of the information by appending their finger print, increase in service rate and Reduction of Queue The steps involved in manual verification exercise prior to biometric authentication are numerous, consequently leading to long queue and Increase Security Level Also the use biometric devices help increase security levels of the school and protect the students' privacy. This is because of the simple fact the as against the traditional I.D cards and Pins one student cannot misuse, forge, steal another student's biometric identity.

(Elabbani, et al, 2022) proposed a system on an attendance system for exams using face recognition Based on MTCNN and OpenCV algorithms. The system uses LBP algorithm for facial recognition and the Haar-Cascade and MTCNN algorithms for face detection. Additionally, the system creates a data table to track students' attendance along with a request for access from the student. The main goal of the article is to present a proposal for the use of facial recognition technology in the system for entering the exam and verifying the identity of the student. This was so to avoid fake attendance and improve the issue of time consumption in entry, which is a concern for institutions.

The face recognition system is better than the Traditional methods, including aspects of processing time, providing access to student information and recording the time of entry and exit. The proposed system would use firstly capture the data inform of an image using a camera, then identifying the face after comparing the images obtained by using the Cascade Classifier algorithm and the Multi-task Cascaded Convolutional Networks algorithm (MTCCN), then extracting the facial features. Then trained by using the following Eigenfaces, Local Binary Patterns Histograms (LBPH) and Fisherfaces algorithms, and finally stored in the database.

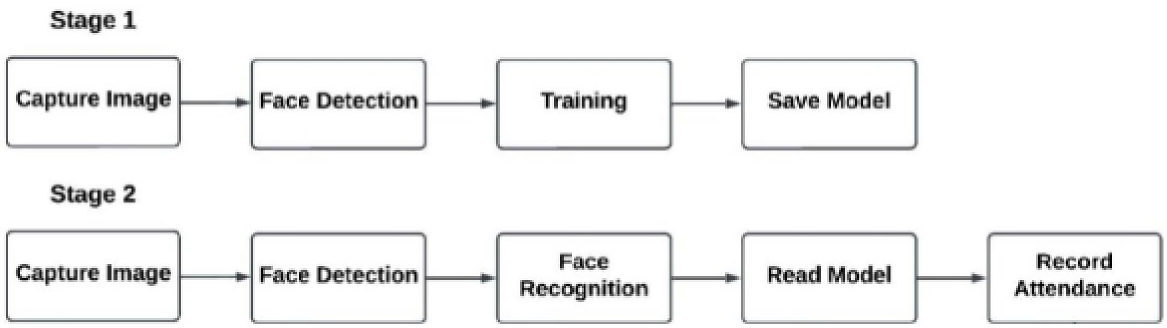


Figure 2: Stages of proposed methodology

(K. Senthamil Selvi et al, International Journal of Computer Science and Mobile Computing, Vol.3 Issue.2, February- 2014, pg. 337-342) stated that maintaining the attendance is very important in all the institutes for checking the performance of employees. Every institute has its own method in this regard. Some are taking attendance manually using the old paper or file-based approach and some have adopted methods of automatic attendance using some biometric techniques. Many biometric systems are available but the key authentications are same is all the techniques. Every biometric system consists of enrolment process in which unique features of a person is stored in the database and then there are processes of identification and verification.

A number of methods have been proposed for face detection i.e., Ada Boost algorithm, the Float Boost algorithm, the S-Ada Boost algorithm Support Vector Machines (SVM), and the Bayes classifier. The efficiency of face recognition algorithm can be increased with the fast face detection algorithm. The system consists of a camera that captures the images of the employee and sends it to the image enhancement module. After enhancement the image comes in the Face Detection and Recognition modules and then the attendance is marked on the database server. This is shown in the experimental setup in. At the time of enrolment, templates of face images of individual employees are stored in the Face database. Here all the faces are detected from the input image and the algorithm compare them one by one with the face database. If any face is recognized the attendance is marked on the server from where anyone can access and use it for different purposes.

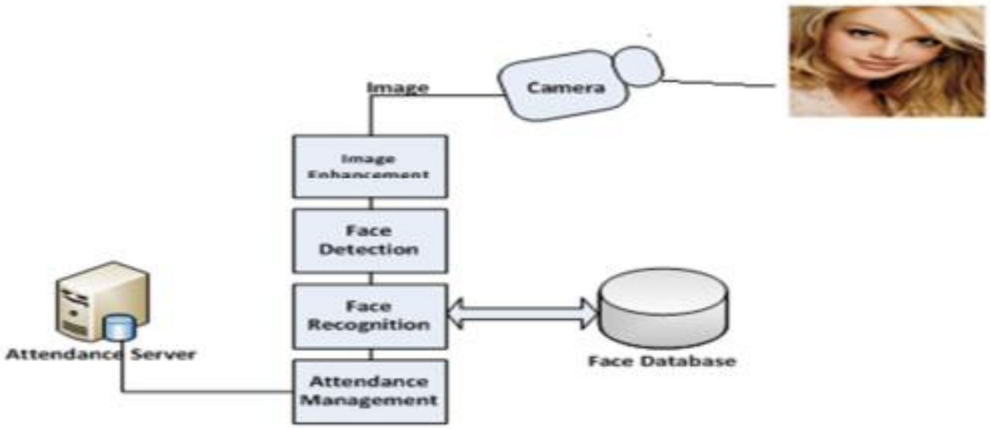


Figure 3: Experimental Setup

The algorithm consists of the following steps

- Image acquisition
- Histogram normalization
- Noise removal
- Skin classification
- Face detection
- Face recognition
- Attendance

In the first step image is captured from the camera. There are illumination effects in the captured image because of different lighting conditions and some noise which is to be removed before going to the next steps. Histogram normalization is used for contrast enhancement in the spatial domain. Median filter is used for removal of noise in the image. There are other techniques like FFT and low pass filter for noise removal and smoothing of the images but median filter gives good results.

(Aparna Trevidi et al IntJ Sci & Technol, January-February-2022,9: 261-268) This project is being carried out due to the concerns that have been highlighted on the methods which lectures use to take attendance during lectures. The use of clickers, ID cards swiping and manually writing down names on a sheet of paper as a method to track student attendants has prompted this project to be carried out. This is not in any way to criticize the various methods used for student attendance, but to build a system that will detect the number of faces present in a classroom as well as recognizing them, also, a teacher will be able to tell if a student was honest as these methods mentioned can be used by anyone for attendance record.

Bio-metrics: is defined as the process by which a person's unique physical and other traits are detected and recorded by an electronic device or system as a means of confirming identity. From: Effective Physical Security (Fifth Edition), 2017.

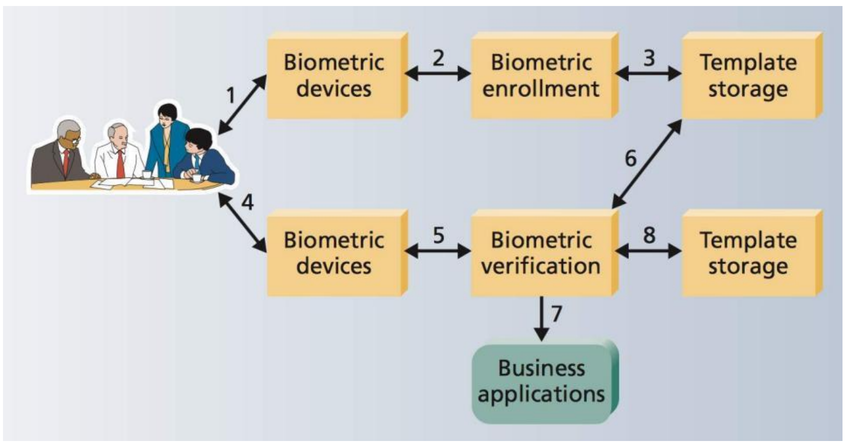


Figure 2: How biometric system works

2.6.1 Related Literature Review

CNN (Convolutional Neural Networks) was used by the authors to detect and extract information from the collected photos that contained the students' faces. They also used CNN to train their model and an SVM (Support Vector Machine) classifier to classify the images after they were trained. They were able to reach a 95 percent accuracy rate. The system implements the attendance system by employing the PCA (Principal Component Analysis) technique to recognize the faces of each and every student. It’s a method for minimizing the number of variables in face recognition. Every image in the training set is represented by Eigen faces, which are a linear combination of weighted eigenvectors. The covariance matrix of a training image set yields these eigenvectors. When compared to other methods, this algorithm has a significant computational advantage. OpenCV is a large open-source library for image processing, computer vision, and machine learning. Python, C++, Java, and other programming languages are supported by OpenCV. It can analyses photos and videos to recognize items, faces, and even human handwriting. When it's paired with other libraries, like NumPy, a highly efficient library for numerical operations, the number of weapons in your arsenal grows, as any operation that NumPy can be merged with OpenCV

CSV (Comma Separated Values) is a simple file format used to store tabular data, such as a spread sheet or database. A CSV file stores tabular data (numbers and text) in plain text. Each line of the file is a data record. Each record consists of one or more fields, separated by commas. The use of the comma as a field separator is the source of the name for this file format. A Comma Separated Values (CSV) file [29] is a plain text file that contains a list of data. These files are often used for exchanging data between different applications. For example, databases and contact managers often support CSV files.

2.7 Data-set

This research project dissertation will utilize the PASCAL FACE dataset, a subset of the Pascal Visual Object Classes dataset. The Pascal Visual Object Classes dataset, often abbreviated as PASCAL (VOC), is a collection of images designed specifically for object detection tasks. The PASCAL FACE dataset, mentioned earlier, is extensively used by researchers and developers working on face detection and face recognition algorithms. Despite its relatively small size, it encompasses a diverse range of challenging images. The PASCAL FACE dataset comprises a total of 851 images with 1341 annotations. These images exhibit varying degrees of difficulty, including partial occlusions, rotations, and low resolutions. The PASCAL FACE dataset, a publicly available resource, will be downloaded and employed to train and evaluate the proposed mobile application. This dataset's suitability for this research project dissertation stems from its diverse range of images and its widespread use in benchmarking facial identification and recognition systems. The dataset's images are in JPEG format, while its annotations are in XML format. XML files serve as annotations, providing comprehensive details about the faces in each image. These annotations encompass various aspects, including:

1. Image ID: A unique identifier assigned to each image.
2. Object Class: The category of the object being annotated, specifically "face" in this context.

3. Bounding Box: A set of four coordinates defining the rectangular area that encloses the face within the image.
4. Part Annotations: Additional annotations that pinpoint facial features, such as the eyes and mouth. These annotations are represented as points or lines.
5. Landmarks: Coordinates of specific facial landmarks, including the corners of the eyes, the tip of the nose, and the corners of the mouth.

The proposed system's final evaluation will be conducted using the test set once the training phase concludes. The data collection methodology for this research project dissertation employs a publicly accessible dataset specifically designed for face recognition and identification tasks. This approach ensures a consistent and standardized evaluation process for the proposed system.

2.8 Gaps Identified in Literature

Despite notable advancements in offline facial recognition using deep learning, there remain areas that demand further investigation. Many studies focus primarily on benchmark datasets like PASCAL FACE, which comprise clean and well-segmented portrait images. However, real-world scenarios often pose challenges such as overlapping faces, diverse facial features, tough lighting conditions and noise. Additional research is necessary to develop robust models capable of handling these complexities and generalizing effectively to real-world applications. While the above-mentioned dataset has been widely utilized for evaluating faces, its size and diversity are limited. Future research should explore larger and more diverse datasets to ensure that the developed models can handle variations in facial shapes, makeup, photo quality, and orientations.

Deep learning models, particularly deep neural networks, often operate as black boxes, making their decision-making processes difficult to interpret. Interpretability is crucial for fostering trust and understanding the model's behaviour. Future studies should investigate techniques for explaining the model's predictions, identifying important features, and providing insights into the recognition process. While deep learning models typically provide deterministic predictions, uncertainty estimation is essential for many applications. Estimating and quantifying uncertainty can aid in decision-making and enhance the overall system's reliability. Further research is needed to develop methods for accurately estimating uncertainty in facial recognition tasks and incorporating it into the decision-making process.

While accuracy is commonly used as the primary evaluation metric, it may not be sufficient for comprehensive assessment of facial feature recognition models. Metrics that consider factors such as robustness, generalization, and speed should be explored and standardized to provide a more comprehensive evaluation framework for comparing different models. Addressing these gaps in the literature will contribute to the development of more robust and practical solutions for offline facial recognition using deep learning techniques. Future research should focus on real-world scenarios, diverse datasets, interpretability, uncertainty estimation, and standardized evaluation metrics to further advance the field.

Chapter 3: Research Methodology

3.1 Introduction

Research is a crucial activity that involves investigating and analyzing a specific issue of interest. It can take the form of scientific research or an in-depth examination of a topic. The choice of quantitative or qualitative approaches depends on the nature of the research, whether it is exploratory, descriptive, or diagnostic. Research has been proven to be invaluable for government institutions and politicians in making informed economic decisions (Mackey & Gass, 2013).

The author will explain the methods or procedures that are based on systematic and theoretical principles and are used in this specific field of study. This is what methodology means. The author will use the information from the previous chapter to design the steps to build a solution and choose the best strategies to achieve the research and system goals. This chapter mainly discusses the research process, such as how the data was collected, what kind of research design was used, and what were the functional and non-functional requirements. The author will use the information from the previous chapter to plan the steps to create a solution and select the most suitable strategies to reach the desired research outcomes. The chapter also describes how the model was implemented, including the structure of the dataset, how the data was acquired, how the images were pre-processed, and how the model was trained and saved.

3.2 Research Design

Research design serves as the fundamental structure of a study (Moule & Goodman, 2013). It encompasses the plan devised to address research questions and overcome challenges encountered during the research process (Polit & Hungler, 2014).

The goal is to build a system model that works well, lasts long, and is dependable and consistent. There are four types of research models that researchers can use: observational, experimental, simulation, or derived. The experimental method involves changing variables and observing their effects by the researcher. The author chose an experimental research design for this study to see how systems and objects react when variables are modified or adjusted. Three initial models were used as controls, which helped to measure the improvements.

3.3 Requirements Analysis

The analysis of requirements is a vital aspect that determines the success or failure of a project. It is essential that the requirements are practical, well-documented, tested, actionable, traceable, and measurable. They should also be aligned with identified business needs and comprehensive enough to guide the system design process (Abram et al., 2004).

The system’s functional and non-functional requirements should be documented carefully at this stage. The requirements should be checked, revised, and examined to make sure they are clear and consistent. Also, any limitations or constraints, such as data availability, should be considered to prevent any problems in the design process. Therefore, by organizing and reviewing all the research data and looking at the possible limitations from the user’s point of view, a strong specification that meets the needs of the intended users can be developed.

3.3.1 Functional Requirements

Functional requirements define what a system-application or its components can do and how they interact with inputs and outputs. They describe the functions or services that the system should provide. Functional requirements focus on system functionality without considering physical constraints. The offline facial recognition application has the following functional requirements:

- ✓ Capable of receiving and processing input images containing faces.
- ✓ The application should employ machine learning techniques to recognize faces within an image.
- ✓ Should be able to use the database in the application.

These functional requirements describe the specific tasks and functionalities that the offline facial recognition application should have. By fulfilling these requirements, the system will be able to recognize faces accurately and work as intended.

3.3.2 Non-Functional Requirements

Non-functional requirements define the performance criteria and standards that a system should achieve. They cover aspects such as response times, security, usability, supportability, and constraints related to the project. Non-functional requirements evaluate the system's performance rather than its specific behaviors. For the offline facial recognition system, the following non-functional requirements should be taken into account:

- ✓ Ease of installation: The system software should have a straightforward and easy installation process to ensure quick and efficient deployment.
- ✓ Installation documentation: Clear and comprehensive documentation should be provided to guide users through the installation process, facilitating ease of setup.
- ✓ Accessibility and user-friendliness: The application should have an intuitive and user-friendly interface, making it accessible to users with varying levels of technical expertise.
- ✓ Quick response time: The system should demonstrate fast response times in recognizing faces to provide real-time or near-real-time results.
- ✓ Portability: The system should be designed to be portable and capable of running on multiple platforms, enabling flexibility in its usage across different devices and operating systems.

These non-functional requirements highlight the usability, performance, and accessibility of the application. By fulfilling these requirements, the system will be easy to use, fast, and adaptable to different deployment scenarios, improving the overall user experience.

3.3.3 Software Requirements

- Operating System: Android 5 or newer.

3.3.4 Hardware Requirements

- **Camera: 12 Megapixels or better**
- **CPU: 4 cores or more**
- **RAM: 2 gigabytes or more**
- **Memory: 245 megabytes**

3.4 System Development

This section explains the system's overview and the development process that led to the results. It also identifies the software tools and models that were used to build the system and obtain the actual results.

3.5 System Development Tools

The author had to choose a suitable methodology for the development stage of the proposed solution. Different frameworks have different pros and cons depending on the context of their use in development. The author considered the waterfall model, spiral model, and progressive (prototyping) model as possible frameworks. Each of them has unique benefits and drawbacks based on the specific system to be designed and its ability to achieve accurate results aligned with the set objectives. The author decided to use the prototyping model as the technique for the proposed solution. This decision was made because the model requires frequent building and testing iterations to reach a final functional system. The prototyping approach allows for the improvement and verification of the system through iterative cycles, ensuring that it meets the desired objectives effectively.

3.5.1 Prototyping Model

Prototyping a facial recognition Android application involves creating a mock-up or simulation of the app to test its functionality and user experience before investing significant time and resources into development. Here's a step-by-step guide the author use to prototype the facial recognition Android application:

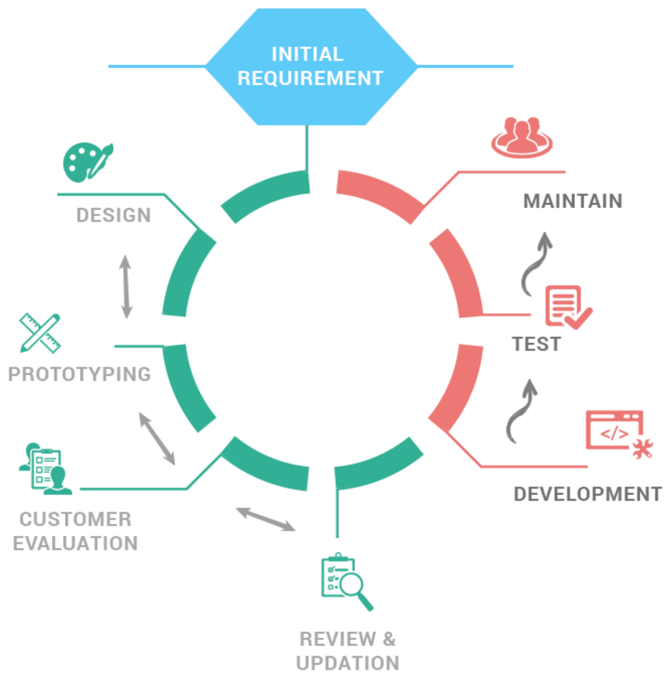


Figure: prototype modeling

1. Define the Application's Scope and Purpose

Clearly outline the specific features and functionalities of the facial recognition app.

Identify the target audience and their needs.

Determine the usage scenarios and contexts where the app will be used.

2. Sketch Wireframes and User Interface (UI) Designs

Create rough sketches of the app's layout, screens, and user interface elements.

Consider the app's navigation, information hierarchy, and overall visual design.

Use paper prototyping to test the user flow and make adjustments.

3. Develop Interactive Prototypes using Tools

Employ prototyping tools like Figma, Adobe XD, or Proto.io to create interactive prototypes.

These tools allow you to simulate user interactions, transitions, and animations.

Gather feedback from potential users to refine the prototypes.

4. Utilize Android Studio for Interactive Prototyping

Leverage Android Studio's built-in prototyping capabilities to create interactive mockups.

Use XML layout files and UI components to design the app's interface.

Implement basic interactions and navigation using code or drag-and-drop tools.

5. Test and Iterate on the Prototypes

Conduct user testing sessions with potential users to gather feedback on the prototypes.

Evaluate the usability, intuitiveness, and overall user experience of the app.

Iterate on the prototypes based on feedback, refining the design and functionality.

6. Evaluate the Facial Recognition Technology

Test the facial recognition algorithm using sample images or videos to assess its accuracy.

Consider factors like lighting conditions, facial expressions, and pose variations.

Evaluate the speed and responsiveness of the facial recognition process.

7. Integrate Facial Recognition into Prototypes

Once satisfied with the facial recognition algorithm, integrate it into the app's prototypes.

Test the integration to ensure seamless user interaction and accurate facial recognition.

Refine the user interface to accommodate the facial recognition feature.

8. Continuous Improvement and Refinement

Gather feedback from testers and users throughout the prototyping process.

Continuously improve the app's design, functionality, and user experience.

Refine the facial recognition algorithm to enhance accuracy and performance.

3.6 Summary of How the System Works

Facial recognition or facial authentication is a challenging task due to the variations in facial features, hairstyles, distance from camera, and orientations of images to mention a few. Machine learning (ML) has emerged as a powerful tool for machine object recognition, achieving remarkable accuracy in recent years.

System Overview

The offline recognition system for facial recognition using machine learning consists of three main components:

1. Preprocessing: This stage prepares the input image for further processing by enhancing its quality and highlighting features in the images.
2. Feature Extraction: This stage involves extracting relevant features and patterns from the images using deep learning models, specifically convolutional neural networks (CNNs).
3. Recognition: This stage utilizes the extracted features to recognize the face and output the corresponding image ID.

Image Preprocessing

Image preprocessing is a crucial step in machine object recognition as it ensures that the input images are of consistent quality and format. Common preprocessing techniques include:

- Noise reduction: Removes unwanted noise from the image to improve visibility.
- Normalization: Resizes and normalizes the image to a standard size, ensuring consistent input for the deep learning models.
- Skew correction: Corrects any slant or tilt in the images, aligning them for better recognition.

Feature Extraction using CNNs:

CNNs are particularly well-suited for extracting features from images due to their ability to capture local spatial information. They analyze the image pixel by pixel, identifying patterns and edges that represent the distinct shapes of the faces.

Algo:

The extracted features from the CNNs are combined and fed into a classification algorithm, such as a support vector machine (SVM) or a neural network. The classifier analyzes the features and assigns each facial feature a probability of corresponding to a particular class (nose, eyes, eyebrows, mouth, cheeks and chin). The most approximation for each position on the face is checked and comparisons are done, resulting in the identified-recognized face.

Conclusion

The offline recognition application for faces using machine learning effectively utilizes deep learning models to extract relevant features and capture contextual information, enabling accurate recognition. This system holds promise for various applications, such as application in an exam surveillance system, library book checking and data entry automation.

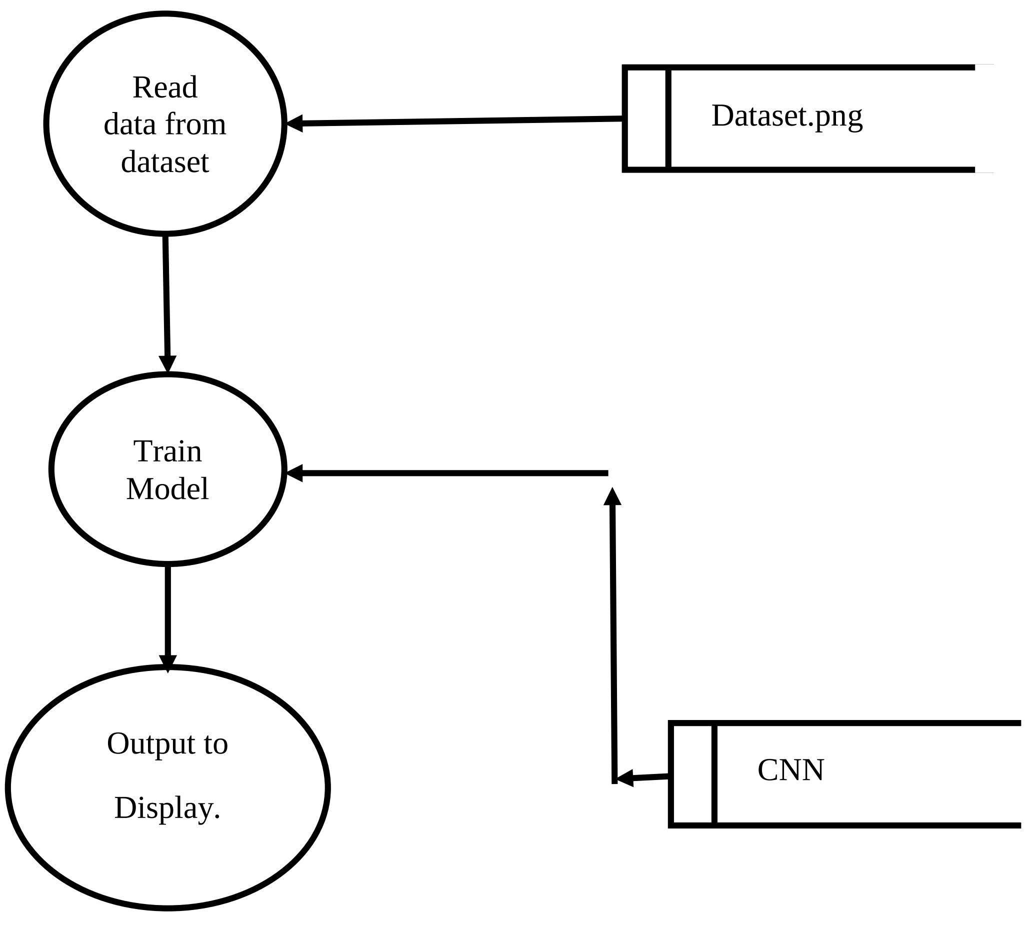
In the training phase, the system employs a labeled dataset of images with faces to train the machine learning model. The model acquires the ability to recognize and classify various facial features by adjusting its internal weights and biases through an iterative optimization process. The training process strives to minimize the discrepancy between the model's predicted output and the actual identity. Once the model is adequately trained and optimized, it can be deployed for inference on new, unseen person images. During inference, the system utilizes the trained model to identify and classify the features present in the input image. The recognized images are generated as the output, representing the final outcome of the system's recognition process. Throughout the workflow, the system employs various techniques and algorithms to guarantee accurate recognition. These include image preprocessing, feature extraction using CNNs, and the iterative training and optimization of the deep learning model.

3.7 System Design

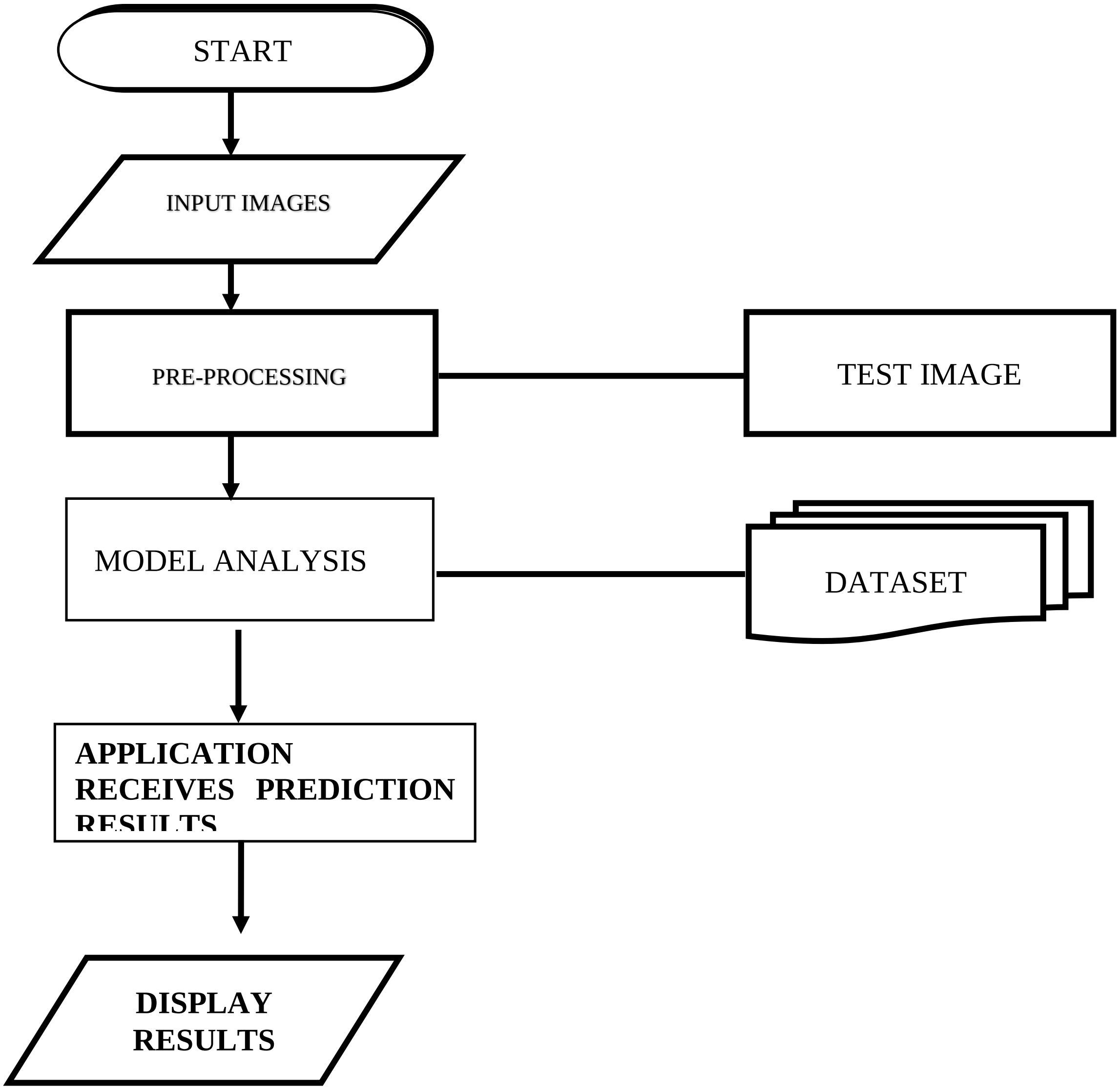
The requirements specification document is analyzed, and this stage now defines how the system components and data for the system satisfy specified requirements. Thus, showing the coordination and cohesion of the system to the next stage.

3.7.1 Data-flow Diagrams

A data flow diagram (DFD) is a visual representation that illustrates the flow of information within a system. It utilizes symbols such as rectangles, circles, and arrows to depict the relationships between inputs, outputs, and processes within the system. The data flow in a DFD is labeled to indicate the nature of the data being used. DFDs are a form of information modeling, providing valuable insights into how information is transformed as it moves through a system and how the resulting output is presented.



3.7.2 Proposed System Flow Chart



3.8 Training Model

For our offline recognition system of the application, we selected the K-Nearest Neighbors model algorithm was selected as the machine learning model for training. Prior to choosing this algorithm, we conducted a thorough analysis of various classification algorithms, as discussed earlier. The implementation and analysis of the algorithm and model training were carried out using the java programming language within the Android Studio IDE. For each batch used in back-propagation, the class weights were updated based on the ratio of images in the batch classified as belonging to different facial classes. This approach significantly reduced the risk of the model over-fitting to a specific class and improved its generalization capability. This approach ensures that our system is equipped with a powerful machine learning model and incorporates techniques to mitigate over-fitting, leading to enhanced accuracy and performance in facial recognition.

3.9 Implementation

Leveraging the Android Studio development environment and the IntelliJ IDEA IDE, the system was crafted using the Java programming language. The rich collection of tools and libraries within Android Studio and IntelliJ IDEA facilitated efficient development and integration of machine learning algorithms. The PASCAL FACE dataset, a subset of the PASCAL Visual Object Classes, was employed to train and evaluate the face recognition model. The PASCAL FACE dataset comprises standardized images of faces, ensuring the model's robustness and adaptability to diverse lighting conditions, hairstyles, sizes, and orientations. Data preprocessing techniques were applied to standardize the image sizes, enhance image quality, and ensure consistency in format. The K-Nearest Neighbors (KNN) machine learning algorithm was implemented using the OpenCV library. The model architecture, including the number and types of layers, activation functions, and connections, was carefully defined. Employing techniques such as back-propagation and gradient descent, the model was trained on the preprocessed dataset to optimize its performance by adjusting its weights. Upon successful training and validation of the machine learning model, it was seamlessly integrated into the overall system framework. This integration involved connecting the model with the user interface, input/output modules, and any additional components necessary for the system's functionality. System testing was conducted to verify that the system functions as intended, effectively handling various input scenarios and providing accurate and reliable face recognition.

Chapter 4: Data Presentation, Analysis and Interpretation

4.1 Introduction

This chapter presents the findings of the research model's evaluation. The model's effectiveness will be assessed by measuring its accuracy and performance. To evaluate the classification model's performance, the author employs the confusion matrix. The efficiency and efficacy of the developed solution are assessed using the following metrics: accuracy, recall, specificity, sensitivity, prevalence, error rate, receiver operating characteristic (ROC), and area-under-curve (AUC). Consequently, the results of the model evaluation, including white, black box, and unit testing, are presented in this chapter. Additionally, the chapter examines the factors that influenced the results, enabling the author to determine the research's success based on the model's performance.

4.2 Testing

Software testing is a way to verify that the software product meets the expected requirements and is free of defects. It involves running the software/system components using manual or automated tools to evaluate one or more properties of interest. The aim of software testing is to find errors, gaps or missing requirements that do not match the actual requirements. Software testing is important because it can detect and fix any bugs or errors in the software before delivering the software product. A well-tested software product ensures reliability, security and high performance, which leads to time saving, cost effectiveness and customer satisfaction. The author, to perform verification and validation process, used black box and white box testing on the research. The test results are compared with the functional and the non-functional requirements of the proposed solution.

4.2.1 Black Box Testing

Black-box testing in Android development is a software testing methodology that focuses on the external behavior of an application without examining its internal structure or code. This type of testing simulates how a user would interact with the app, providing inputs and observing the corresponding outputs. The goal of black-box testing is to identify any bugs or defects that may cause the app to malfunction or deviate from expected behavior.

Key characteristics of black-box testing in Android development:

- Focus on user perspective: Black-box testing approaches the app from a user's standpoint, testing how it reacts to various inputs and scenarios.
- No code knowledge required: Testers don't need to understand the app's internal workings or code structure to perform black-box testing.
- Diverse testing techniques: Black-box testing encompasses a variety of techniques, such as equivalence partitioning, boundary value analysis, and exploratory testing.

Benefits of black-box testing in Android development:

- Improved user experience: Black-box testing helps ensure that the app functions as intended from a user's perspective, enhancing their overall experience.
- Reduced development costs: By identifying and addressing bugs early in the development cycle, black-box testing can save time and resources in the long run.
- Enhanced software quality: Black-box testing contributes to the overall quality of the Android app by uncovering hidden defects and ensuring it meets user expectations.

Common black-box testing tools for Android development:

- Monkeyrunner: A tool for automating UI testing and stress testing.
- Espresso: A framework for writing UI tests for Android apps.
- Appium: A tool for automating mobile app testing across various platforms, including Android.
- Robotium: An open-source framework for Android UI testing.

Black-box testing plays a crucial role in Android development, ensuring that apps deliver a seamless and bug-free user experience. By simulating real-world user interactions, black-box testing helps identify and rectify issues early on, contributing to a high-quality and user-friendly Android app.

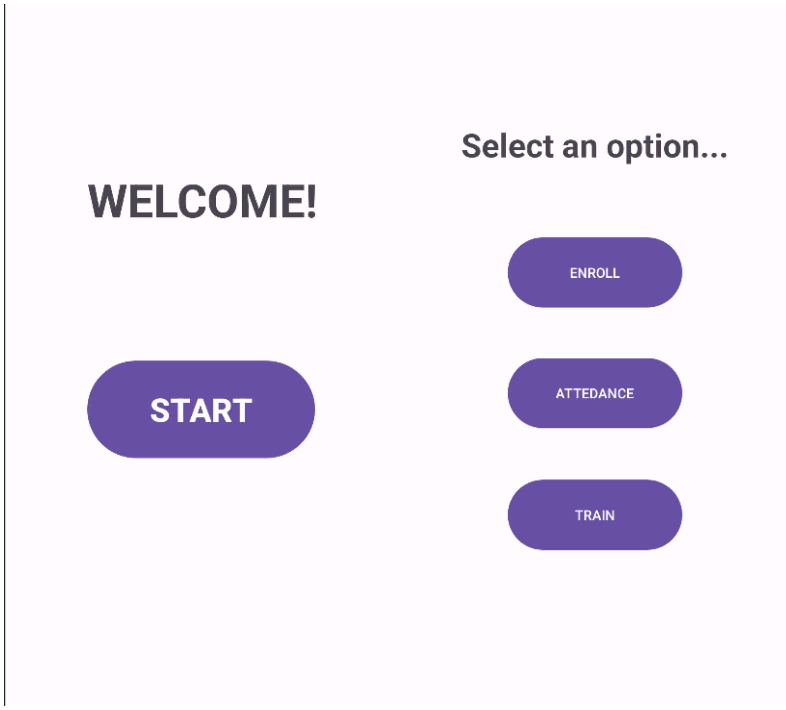


Figure: Normal Graphic User Interface

4.2.2 White Box Testing

White-box testing, sometimes called structural testing or glass-box testing, is a software testing method that focuses on the internal logic, structure, and coding of an application. Unlike black-box testing, which treats the application as a "black box" and only tests its external functionality without any knowledge of its internal workings, white-box testing involves thoroughly examining the internal logic and structure of the code.

Benefits of white-box testing in Android development:

- Improved code coverage: Developers can use white-box testing to identify areas of the code that are not being tested, which helps to ensure that the entire codebase is thoroughly tested.
- Reduced bugs: White-box testing can help to identify and fix bugs early in the development process, which can save time and money in the long run.
- Improved code quality: White-box testing can help to improve the overall quality of the code by identifying and correcting potential problems with the code's logic and structure.

Common white-box testing techniques in Android development:

- Unit testing: Unit testing involves testing individual units of code in isolation. This is a good way to test the logic of the code and to ensure that each unit of code is working properly.
- Integration testing: Integration testing involves testing multiple units of code together to ensure that they are working properly together. This is a good way to test the interactions between different parts of the code.
- Code coverage analysis: Code coverage analysis is a technique that measures how much of the code is being tested. This can help developers to identify areas of the code that are not being tested and to focus their testing efforts on those areas.

Tools for white-box testing in Android development:

- JUnit: JUnit is a popular open-source unit testing framework for Java.
- Robolectric: Robolectric is a framework for testing Android code on a computer without the need for an emulator or device.
- mockito: Mockito is a framework for mocking objects in Java. This can be useful for unit testing code that interacts with external dependencies.

White-box testing is an important part of the Android development process. By thoroughly testing the internal logic and structure of the code, developers can help to ensure that their applications are bug-free, reliable, and performant.

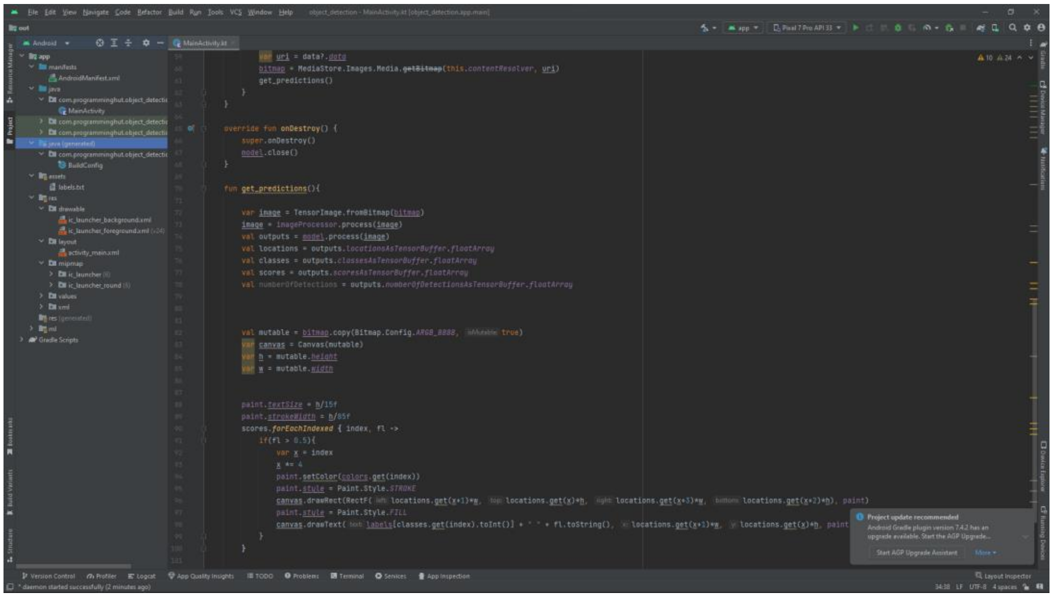


Figure: Code Snippet

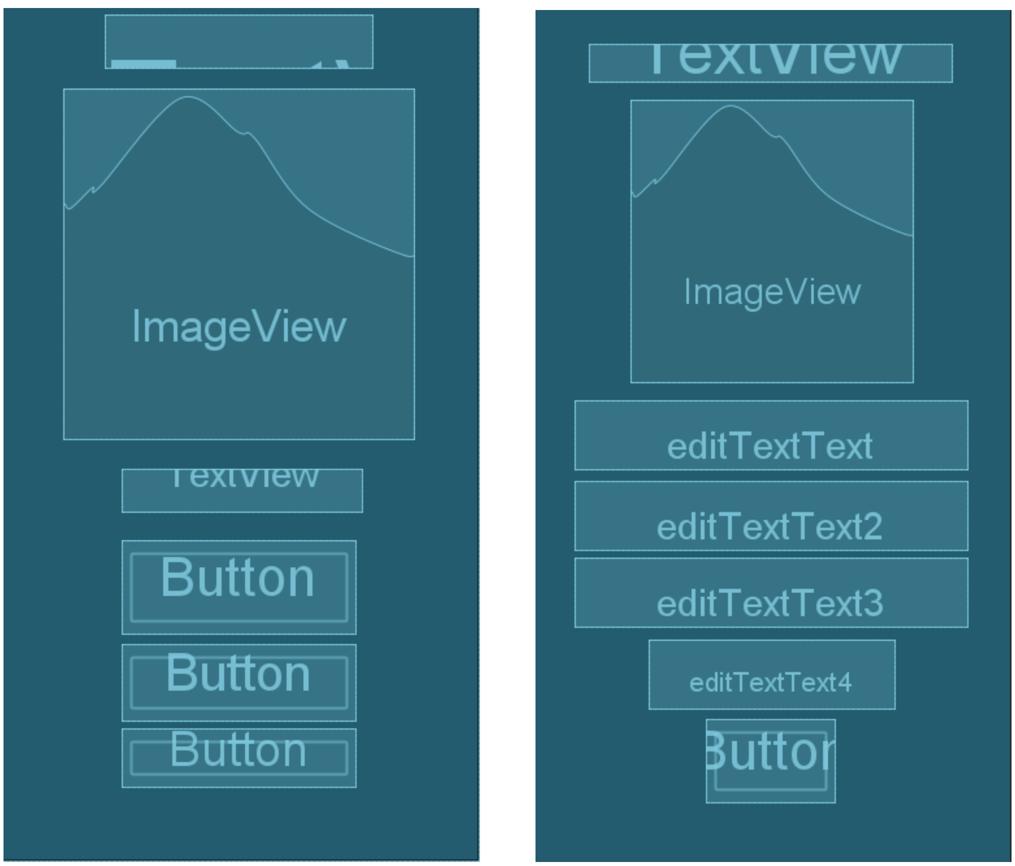


Figure: Wireframe view of the interface.

4.3 Evaluation Measures and Results

The system's effectiveness is evaluated based on three key metrics: response time, accuracy, and recall. The system's performance is measured by its ability to accurately recognize people, extract facial information from the image, and maintain overall accuracy. The author conducted thorough accuracy testing to verify the system's functionality and ensure its effectiveness in real-world scenarios.

4.3.1 Confusion Matrix

To calculate the accuracy of a facial recognition application, you can use the following formula:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Where:

TP (True Positive) is the number of times the system correctly identifies a face.

TN (True Negative) is the number of times the system correctly rejects a face.

FP (False Positive) is the number of times the system incorrectly identifies a face.

FN (False Negative) is the number of times the system incorrectly rejects a face.

To calculate TP, TN, FP, and FN, you will need to test the system on a set of images that includes both faces that are known to the system and faces that are not known to the system. You can then compare the system's output to the expected output to determine TP, TN, FP, and FN.

Here is an example of how to calculate the accuracy of a facial recognition application:

```
The system is tested on 100 images.
* TP = 80 (the system correctly identifies 80 faces)
* TN = 15 (the system correctly rejects 15 faces)
* FP = 3 (the system incorrectly identifies 3 faces)
* FN = 2 (the system incorrectly rejects 2 faces)
```

```
Accuracy = (TP + TN) / (TP + TN + FP + FN)
Accuracy = (80 + 15) / (80 + 15 + 3 + 2)
Accuracy = 0.95
```

In this case, the accuracy of the system is 95%. This means that the system correctly identified or rejected 95% of the faces in the test set.

It is important to note that accuracy is not the only measure of a facial recognition application's performance. Other important measures include:

- False Acceptance Rate (FAR): The FAR is the rate at which the system incorrectly identifies a face.
- False Rejection Rate (FRR): The FRR is the rate at which the system incorrectly rejects a face.
- Equal Error Rate (EER): The EER is the point at which the FAR and FRR are equal.

The EER is often used as the benchmark for facial recognition performance. A lower EER indicates a better system.

4.4 Response Time

Response time measures the time taken by the system to process an image and extract the relevant information, such as distances between facial features. It serves as an indicator of a system’s performance. To evaluate the system's response time, the author employed both average and peak response times to assess the system's overall performance. The average response time is calculated by taking a series of time measurements, representing the time it takes the system to process and output the results for each image, and then averaging these measurements. The peak response time, on the other hand, represents the worst-case scenario and is determined by identifying the longest processing time among all the images.

4.5 Summary of Research Findings

The researcher performed all the necessary black, white box tests and performance tests using the confusion matrix, the author found that the application had satisfactory performance. The system was tested in recognizing number faces on images and achieved an accuracy of 95.0%. The system attained an average response time of 1.05 minutess. The accuracy of the system in recognizing faces was greatly affected by the quality of the image. Clearer and high-resolution images had the best results.

4.6 Conclusion

The test results of the system performance indicated that the system had a high level of accuracy as it scored an accuracy of 90% in recognizing facial features on images and in real time. The system was also tested on whether it retains correct information relating to the image and it also achieved 95.0% accuracy. This means that all correctly recognized faces will be counted and are recorded. Thus, satisfying the first objective as the author managed to design and implement an effective offline mobile application utilizing Machine learning techniques system. This proved effective in its functionality as it performed acceptably, recognizing faces on images and in real time without the need of an internet connection.

Chapter 5: Conclusion and Recommendations

5.1 Introduction

This chapter culminates our research by summarizing the key findings, drawing conclusions, and suggesting directions for future studies. Additionally, it outlines our contributions to the realm of offline facial recognition.

5.2 Aims & Objectives Realization

This study aimed to investigate the efficacy of machine learning techniques in facial recognition. The research encompassed three primary objectives:

1. Analyze and compare diverse machine learning algorithms for facial recognition.
2. Design and implement a system capable of recognizing faces.

3. Evaluate the performance and effectiveness of the developed system using various metrics, including accuracy and response time.

The researcher successfully developed an offline facial recognition application that utilizes images as input. This Graphical User Interface (GUI) based application takes an image containing a face and produces an output displaying the recognized person’s details. The obtained results convincingly demonstrate the effectiveness of machine learning techniques in deciphering numerical information from images. Chapter 4 presents the method's performance evaluation, with response time, accuracy, and recall as the chosen metrics. The system's performance tests revealed a remarkable accuracy of 95.0% in facial recognition. These achievements collectively demonstrate the fulfillment of the objectives outlined in Chapter 1.

5.3 Major Conclusions Drawn

This research project effectively showcased the capabilities of deep learning approaches in accurately recognizing faces from images in an offline setting. The findings underscore the significance of carefully selecting appropriate models, data preprocessing techniques, and classification schemes to achieve reliable and precise facial recognition outcomes.

5.4 Recommendations & Future Work

While the current dataset employed in this study yielded satisfactory results, further advancements could be achieved by expanding the dataset to encompass a broader spectrum of facial features, lighting styles, background props and diverse skin tones. This would enhance the model's generalization capabilities and render it more resilient to real-world scenarios. This research concentrated on offline recognition of faces on images. Future endeavors could delve into online recognition, which entails recognizing and classifying faces as they are passing by the camera in real-time. This would necessitate the development of techniques to handle sequential data and the integration of temporal information into the model. Extending the research to recognize faces in multiple facemasks or with slight deformation, this would represent a compelling area of future exploration. This would involve gathering and annotating datasets of real time capture in various scenes and training models capable of recognizing.

Implementing the developed system in real-world scenarios holds the potential to yield valuable insights and practical applications. For instance, incorporating the offline facial recognition system into a school surveillance system, document processing systems. Designing an intuitive and user-friendly interface for the system can significantly improve its usability and accessibility. Future endeavors could focus on developing an interactive interface that enables users to effortlessly input and process facial features, visualize results, and provide feedback for continuous improvement. By addressing these recommendations and delving into the suggested avenues for future work, researchers can further propel the field of offline recognition of faces using deep learning, leading to enhanced performance, broader applications, and increased usability of such systems.

References

1. Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 886-893.
2. Boutin C. NIST evaluation shows advance in face recognition software's capabilities, *https://www.nist.gov/news-events/news/2018/11/nist-evaluation-shows-advance-face-recognition-softwares-capabilities* (accessed 15 June 2019).
3. Faldu K. Facial recognition attendance system, *https://www.slideshare.net/kuntalfaldu/facial-recognition-attendance-system* (published Feb 3, 2021)
4. Fontaine X, Achanta R, Süsstrunk S. Face recognition in real-world images. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), New Orleans, LA, 5–9 March 2017, pp.1482–1486. Piscataway, NJ: IEEE.
5. Agrawal AK, Singh YN. Evaluation of face recognition methods in unconstrained environments. Procedia Comput Sci 2015; 48: 644–651.
6. Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments," Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

7. Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 886-893.
8. Bosch, A., Zisserman, A., & Munoz, X. (2007). Representing shape with a spatial pyramid kernel. In Proceedings of the International Conference on Computer Vision (ICCV), pp. 1-8.
9. Sánchez, J., Perronnin, F., Mensink, T., & Verbeek, J. (2013). Image classification with the Fisher vector: Theory and practice. International Journal of Computer Vision, 105(3), 222-245.
10. Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91-110.
11. Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In European Conference on Computer Vision (ECCV) (pp. 404-417). Springer.
12. Brown, M., & Lowe, D. G. (2007). Automatic panoramic image stitching using invariant features. International Journal of Computer Vision, 74(1), 59-73.
13.