

UNIQUE ATTENDANCE SYSTEM WITH GROUP MARKING OF THE STUDENTS PRESENCE WITH A TIME SAVING APPROACH

L.Raji Assistant Professor
Department of Computer Science and Engineering
R.M.K. Engineering College Kavaraipettai, India
lri.cse@rmkec.ac.in

Rahul Maddula, Pinupolu Ramanaiah Saiteja, Akshay Kumar T N
IV year, Department of Computer Science and Engineering
R.M.K. Engineering College Kavaraipettai, India
rahu17231.cs@rmkec.ac.in, aksh17102.cs@rmkec.ac.in,
pinu17226.cs@rmkec.ac.in

Abstract- The Attendance Management System is a necessary tool for taking attendance in any environment where attendance is critical. Most of the existing approaches, however, take time, requiring users to work manually. The Automated Management System carries out daily attendance marking and analysis activities with reduced human interventions and develops a mobile application with a single shoot attendance system to catch the student's group of faces and verify that their face is present in the database for this project. A report is generated and sent automatically. To employees' registered postal identity as the attendance period closes. This project, therefore, contributes to saving people's time and automates records that help manual maintenance save.

I. INTRODUCTION

The Attendance Management System is software designed to monitor regular student attendance in schools, universities, and other educational institutions. It allows you to look up a specific student's attendance details in a specific class. The data is sorted by the operators, whom the instructor for each lesson supplies. This device can also be used to assess a student's attendance eligibility. The aim of creating an attendance management system is to modernise the traditional method of taking attendance. Another reason for creating this programme is to automatically produce the report at the end of the session or in its middle.

The system being developed is cost-effective from the standpoint of a school or college. It is cost-effective in the sense that all paperwork has been eliminated. The system is also time-efficient because the calculations are automated and performed at the end of the month or as needed by the user. As the data is required, the obtained results have few errors and are highly accurate. The system's technical requirements are minimal, and it does not require any additional hardware or software. The system working is relatively easy to use and learn due to its simple but attractive interface. The method does not necessitate any additional preparation. For the user to operate. The Present System's Operation In the current system, all work is done on paper. The entire session attendance is saved in a register, and reports are generated at the end of the session. We are not interested in generating reports in the middle of a session or the requirement because it takes more time to calculate. Students that do not have a 75% attendance rate are notified at the end of the session.

The new system is challenging to use since data retrieval is slow and data is not efficiently maintained. We need more calculations to generate the report to do it at the end of the session. Moreover, the student is not given a single opportunity to improve their attendance. Because all calculations to generate the report are done manually, there is a higher chance of error. The current system necessitates a significant amount of paperwork. Even the loss of a single register/record created a problematic situation because all of the papers were required to generate the reports. Because all work is done manually, we cannot generate reports in the

middle of a session or as needed because it is very time-consuming.

Characteristics of Attendance Management System: The proposed system is user-friendly because data retrieval and storage are quick, and data is efficiently maintained. Furthermore, the proposed system includes a graphical user interface, allowing users to interact with the system with ease. The proposed system reports are easily generated, so the user can generate the report as needed (monthly) or during the session. The user can send a notice to the students to encourage them to attend class regularly. The proposed system necessitates very little paperwork. All data is immediately fed into the computer, and reports can be generated using computers.

Furthermore, because there is no need to keep data on paper, work becomes much more accessible. There will be computer operator control, so there will be no room for error. Furthermore, storing and retrieving information is simple. As a result, work can be completed quickly and on time.

II. LITERATURE REVIEW

Jingxiao Zheng, Rajeev Ranjan, Ching-Hui Chen, Jun-Cheng Chen, Carlos D. Castillo, and Rama Chellappa.

Although deep learning methods have outperformed humans in still image-based face recognition, unconstrained video-based face recognition remains a difficult job. E enormous volume of data that needs to be processed and intra/inter-video variations in posture, lighting, occlusion, scene, blur, video quality, and so on, consider challenging scenarios for face recognition from multiple-shot videos and surveillance videos with low-quality frames for unconstrained video-based face recognition in this work. Face/fiducial identification, face association, and face recognition modules make up a robust and practical framework for unconstrained video-based face recognition to handle these problems. To begin, we use single-shot multi-scale face detectors to localize faces in videos efficiently. The detected faces are then grouped face association approaches that are specifically developed for multi-shot images. Finally, based on an unsupervised subspace learning approach and a neural network, the proposed face matcher recognizes the

faces—subspace-to subspace similarity metric. Extensive experiments on challenging video datasets, such as Multiple Biometric Grand Challenge (MBGC), Face and Ocular Challenge Series (FOCS), IARPA Janus Surveillance Video Benchmark (IJB-S) for low-quality surveillance videos and IARPA JANUS Benchmark B (IJB-B) for multiple-shot videos, demonstrate that the proposed system can accurately detect and associate faces from unconstrained videos and effectively learn robust and discriminative features for recognition.

Pros:

Face recognition is much faster in identification compared to other methods.

Cons:

A constant surveillance camera is required for this method to become a reality, thus making this project too expensive to run.

Xiaojuan Cheng, Jiwen Lu, S. Bo Yuan, Member, Jie Zhou, Senior Member.

To exploit face recognition, a Face Segmenter-Enhanced Network (FSENet) was developed for localized facial property. Most existing methods emphasize the holistic characteristics of entire face images, limiting discriminative ability due to significant intra-class variations and inter-class fine-grain. To address this, we present a face segment to parse the face into local components and explore their internal correlations, strengthening the discriminability to discern identities. Specifically, We implement a semantic parsing module that assigns a semantic component mark to each pixel. This module produces parsing maps, each reflecting the pixel-wise probability of a specific event occurring in a particular component of the face. To achieve local features, we segment facial regions masked by the parsing maps. We will continue to enhance the structural correlation of facial component features to improve customized attributes. We finally incorporate holistic and local information to enhance the discriminative power of the face descriptor. Extensive experiments on popular public-domain datasets, including labelled Face in the Wild (LFW), youtube Faces (YTF), IARPA IJB-A, IJB-B and IJB-C, and the MegaFace Challenge show that our method achieves promising performance.

Pros:

This method achieves promising performance.

Cons:

The method only focuses on the performance side and not on the attendance module alone.

iii. EXISTING SYSTEM

Though deep learning approaches have surpassed human success in still image-based face recognition, unconstrained video-based face recognition remains a challenge.

Due to the vast amount of data to be processed and intra/inter-video variations in posture, lighting, occlusion, scene, blur, video quality, and so on, this is a difficult job. We look at challenging scenarios for unconstrained video-based face recognition from multiple-shot videos and surveillance videos with low-quality frames in this paper. We propose a robust and efficient framework for unconstrained video-based face recognition that consists of modules for face/face detection, face association, and face recognition to resolve these issues. To begin, we use multi-scale single-shot face detectors to locate faces in videos efficiently. Faces are

detected and grouped using carefully developed face association methods, beneficial for multi-shot images. Finally, using an unsupervised subspace learning technique and a subspace-to-subspace similarity measure, the proposed face matcher recognises the faces.

EXISTING SYSTEM DISADVANTAGES

- Faces are the only thing that is noticed in a picture or video.
- It is unable to identify the person whose face appears in the picture or video.
- The algorithm used is inefficient, and recognising it takes longer.

IV. PROPOSED WORK

In any situation where attendance is essential, an attendance management system is required for taking attendance.

However, most current methods are time-consuming, invasive, and enable users to do manual work. Face recognition is a crucial application in image processing since it is used in so many different fields. One application of face recognition is the identification of individuals in an organization for attendance. The maintenance and evaluation of attendance records are critical in every organization's performance review. The aim of creating an attendance management system is to automate the conventional method of taking attendance. The Automated Attendance Management System reduces the need for human involvement in the everyday activities of attendance marking and study. In this project, we will create a single-shot attendance system mobile application that will capture a group of students' faces, verify their presence in the database, and mark them as a present. As soon as the attendance period ends, an automated report is produced and sent to the employee's registered email address. As a result, this project saves people's time and automates records, reducing the need for manual maintenance. The introduction of this project also opens up a slew of new possibilities, such as biometric scanning in bank lockers.

BENEFITS:

- This project lets people save time by automating records, which eliminates the need for manual maintenance.
- Automatically sends a note to the teachers, instructing them to keep track of class attendance.
- The entire attendance system is being automated.
- Time well spent in the growth of students
- In real time, it's easy to use.

POSSIBLE APPLICATIONS:

- This is a term that is commonly used in schools and universities.
- MNC companies use it.
- Wherever an attendance system is needed, this device is used.

V. TECHNOLOGIES USED

OpenCV

Gary Brodsky founded OpenCV at Intel in 1999, and the first release was in 2000. Vadim Pisarevsky joined Gary Brodsky to lead Intel's Russian OpenCV software team. OpenCV was used on Stanley, the vehicle that won the 2005 DARPA Grand Challenge in 2005. Later, it continued to be actively developed with Willow Garage's help, with Gary Brodsky and Vadim Pisarevsky leading the project. OpenCV currently supports a

wide range of algorithms related to computer vision and machine learning, and its capabilities are expanding daily. Gary Brodsky founded OpenCV at Intel in 1999, and the first release was in 2000. Vadim Pisarevsky joined Gary Brodsky to lead Intel's Russian OpenCV software team. OpenCV was used on Stanley, the vehicle that won the 2005 DARPA Grand Challenge in 2005. Later, it continued to be actively developed with Willow Garage's help, with Gary Brodsky and Vadim Pisarevsky leading the project. OpenCV currently supports a wide range of algorithms related to computer vision and machine learning, and its capabilities are expanding daily. Because OpenCV is released under the BSD licence, it is free for both academic and commercial use. It supports Windows, Linux, Mac OS, iOS, and Android and has C++, C, Python, and Java interfaces. OpenCV was created with a strong emphasis on real-time applications and computational efficiency in mind. The library, written in optimised C/C++, can take advantage of multi-core processing. OpenCV has been adopted worldwide, with a user community of over 47 thousand people and an estimated number of downloads over 6 million. The applications range from interactive art to mine inspection, map stitching on the web, and advanced robotics.

short period due to its simplicity and code readability. It allows the programmer to express his ideas in fewer lines of code while still maintaining readability. Python is slower than C/C++ and other programming languages. However, another important feature of Python is that it is easily extensible with C/C++. This feature allows us to write computationally intensive code in C/C++ and then wrap it in Python so that we can use it as a Python module. This provides us with two benefits: first, our code is as fast as original C/C++ code (because it is the actual C++ code running in the background), and second, Python is straightforward to code in. This is how OpenCV-Python functions; it is a Python wrapper around the original C++ implementation. Furthermore, Numpy's assistance makes the job easier. Numpy is a numerical operations library that is highly optimised. It generates MATLAB-style syntax. All array structures in OpenCV are converted to and from Numpy arrays. So, whatever operations Numpy can perform, you can combine them with OpenCV, increasing the number of weapons in your arsenal. Aside from that, several other libraries, such as SciPy and Matplotlib, which support Numpy, can be used with this. As a result, OpenCV-Python is an ideal tool for rapid prototyping of computer vision problems.

OpenCV-Python

Guido van Rossum developed Python, a general-purpose programming language. That has grown in popularity in a

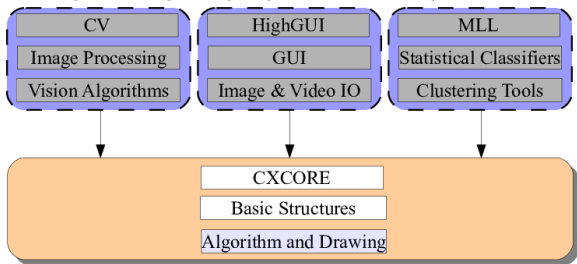


Figure 2.1 OpenCV software architecture

Face detection

So, whatever operations Numpy can perform, you can combine it with OpenCV, increasing the number of weapons in your arsenal. Other libraries, such as SciPy and Matplotlib, which support Numpy, can also be used with this. As a result, OpenCV-Python is an excellent tool for quickly prototyping computer vision problems.

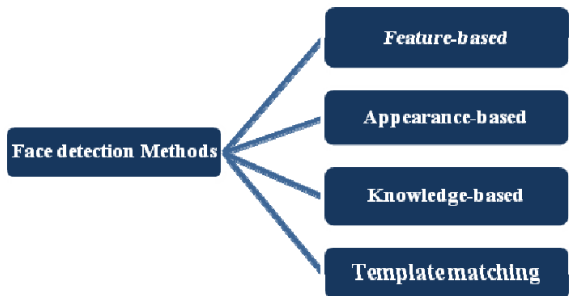


Figure 2.2 Face detection methods

Knowledge-Based

The knowledge-based method relies on a set of rules to detect faces and is based on human knowledge. For example, a face must have a nose, eyes, and mouth that are all within certain

distances and positions from each other. The main issue with these methods is the difficulty in developing a suitable set of rules. If the rules were too broad or too specific, there could be a lot of false positives. This method is insufficient for finding multiple faces in multiple images.

Feature-Based

Faces are located using the feature-based method by extracting structural features from the face. It is trained as a classifier first and then used to distinguish between facial and non-facial regions.

The target is to get out of the bind. Constraints of instinctive face recognition. This method, divided into several steps and even photos with many faces, had a 94 percent success rate.

Template Matching

The Template Matching method locates or detects faces by correlating the templates and input images using pre-defined or parameterized face templates. A human face, for example, can be divided into four parts: eyes, face contour, nose, and mouth. Besides, a face model can be constructed solely from edges using the edge detection method. This method is simple to implement, but it is insufficient for face detection. However, deformable templates have been proposed as a solution to these issues.

Figure 2.4 Geometric face recognition

2. Photometric stereo: This technique is used to reconstruct an object's shape from a series of images taken under various lighting conditions. A gradient map is a set of surface normals used to create a pattern and defines the recovered object's shape.

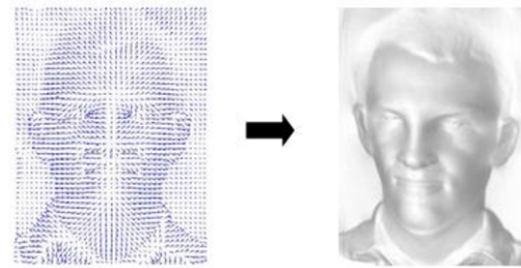


Figure 5.5 Photometric stereo for face recognition

React native app development

Cross-platform development has evolved into an excellent substitute for fully native mobile app advancement. You make different Android and iOS applications using the native mobile development approach. By using the same code on both platforms, cross-platform development allows you to save money and time. The React Native framework is a growing mobile solution widely regarded as the future of cross-platform mobile app development.

React Native is open-source software that is free to use. for building mobile apps. Mobile apps using only JavaScript. Jordan Walke, a Facebook software engineer, introduced it as a new technology for easier development and a better user experience. The key distinguishing feature of this platform is that React Native apps behave just like native apps. They are not different from Java, Objective-C, or Swift apps, and they use the same UI building blocks as native iOS or Android apps. Building a mobile app with React Native, on the other hand, is much simpler and less costly.

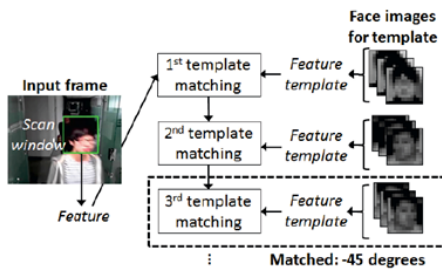


Figure 2.3: Template Matching

Appearance-Based

To find out face models, the appearance-based method relies on a set of delegate training face images. The appearance-based approach outperforms other methods of performance. In general, appearance-based methods rely on statistical analysis and machine learning techniques to determine the relevant characteristics of face images. This method is also used in the extraction of features for face recognition.

Face recognition

Identifying a previously detected object as a known or unknown face is a difficult job, Is known as face recognition. The problems of face recognition and face detection are frequently confused. Face Recognition, on the other hand, determines whether the "face" is known or unknown by using a database of faces to validate this input face.

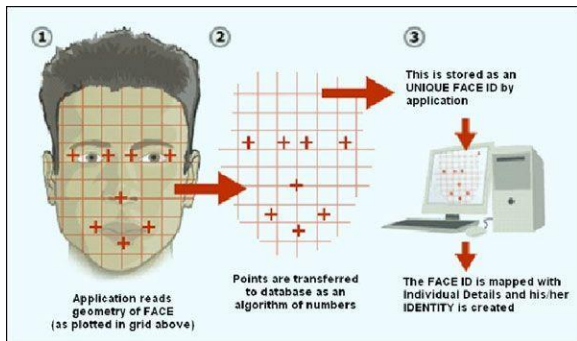
Different approaches to face recognition

Different approaches of face recognition

There are two approaches to the face recognition problem that are widely used:

geometric (feature-based) and photometric (view based). As the interest in face recognition among researchers grew, many different algorithms were developed, three of which have been extensively studied in the face recognition literature. There are two approaches to recognition algorithms:

1. Geometric: This is based on the geometrical relationship of facial landmarks or the spatial configuration of facial features. The most prominent geometrical features of the face, such as the eyes, nose, and mouth, are shown first. Identified, and then faces are classified based on different geometrical distances and angles between features.



VI. SYSTEM DESIGN

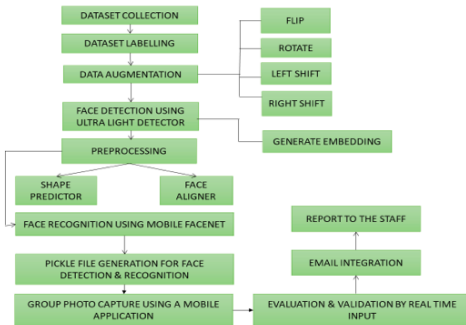


Figure 6.1 : Architecture Diagram

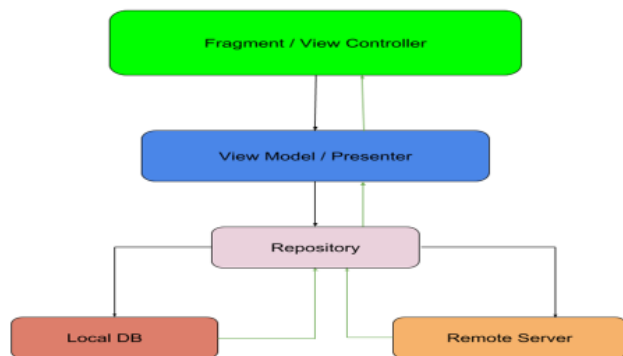


Figure 6.2 : React native Architecture Diagram

WORKING:

To begin, the Intel Processor was set up to work with the Linux operating system. The project's required libraries are installed using the terminal command window, which is also used to run all of the project's commands and run the project. The photographs of the students are used to assemble the datasets for student attendance. The Intel Processor now uses a deep learning algorithm to train the datasets to automatically identify the correct face and mark the student as present or absent. Now, using a mobile application, a group photo of the students is taken, and the faces of the students in the photo are identified and extracted. Face recognition is performed using the MobileFaceNet and Ultralight face detection algorithms. The student is marked present if his or her face matches any of the faces in the database. Thus, it goes until the attendance period is over. When a student is absent, an automatic report of the student's absence is sent to the class teacher via mail. As a result, this project assists in successfully implementing an automatic attendance system using the most up-to-date face recognition technology while also saving the institution time.

Modules Description:

- Dataset collection
- Data augmentation
- Face detection module
- Face Recognition Module
- Email Integration
- Mobile app development

Dataset Collection:

A data set is a group of data. Deep Learning has emerged as the preferred technique for tackling a wide range of complex real-world problems. It is, without a doubt, the most effective technique for computer vision tasks. Deep learning's power in computer vision is demonstrated in the picture above. A deep network will segment and recognise the "key points" of every person in an image with enough training. These deep learning machines, which have been performing admirably, need much fuel, which is data. Our model performs better when there is more labelled data available. Google has even experimented with the idea of more data leading to better performance on a large scale, with a dataset of 300 million images. When using a Deep Learning model in a real-world application, it must be fed data regularly to improve its performance. Data is, without a doubt, the most precious resource in the deep learning era. The data collection process is divided into three steps.

Scraping From the Web:

Because of the amount of human work involved, manually locating and downloading images takes a long time. The task most likely requires the detection of familiar objects. As a result, the term "web scraping" is coined. It also becomes the object's class name. Every Pixel in the image must be used. It is best to use some of the many excellent image annotation tools that are already available. Can create pixel labels for segmentation given a rough set of polygon points around an object. The extreme deep cut is similar to the extreme deep cut, except that only the four extreme points around the object are used. This will result in some good segmentation and bounding box labels. Another alternative is to use an image annotation GUI that already exists.

Third-party:

Because data has become such a valuable commodity in the deep learning era, many start-ups have begun to offer their image annotation services, where they will collect and label data—describing the data and annotations required. Mighty, a company that specialises in self-driving car image annotation and has grown to be a significant player in the field, was also present at CVPR 2018. Payment AI is less specialised than Mighty AI, and it can annotate images from any domain.

Data Augmentation:

The amount of data available also increases the performance of deep learning neural networks.

Data augmentation is a method of artificially creating new training data from existing data. This is accomplished by using domain-specific methods to transform examples from the training data into new and unique training examples.

The most well-known type of data augmentation is image data augmentation, which entails transforming images in the training dataset into transformed versions that belong to the same class as the original image.

Shifts, flips, zooms, and other operations from image manipulation are included in transforms.

The aim is to add new, plausible examples to the training dataset. This refers to variants of the training set images that the model is likely to see. A horizontal flip of a cat photo, for example, would make sense because the photo could have been taken from either the left or right. A cat photo's vertical flip makes no sense and would almost certainly be inappropriate, given that the model is unlikely to see an upside-down cat photo.

Convolutional neural networks(CNN), for example, are modern deep learning algorithms, Can learn independent features of where they appear in the picture. However, augmentation can help with this transform invariant approach to learning by assisting the model in learning characteristics that are also transforming invariant, such as left-to-right to top-to-bottom ordering, light levels pictures, and so on.

Typically, image data augmentation is only used on the training dataset, not the validation or test datasets. Image resizing and pixel scaling are examples of data preparation, which differs in that it must be done continuously across all datasets that interact with the model.

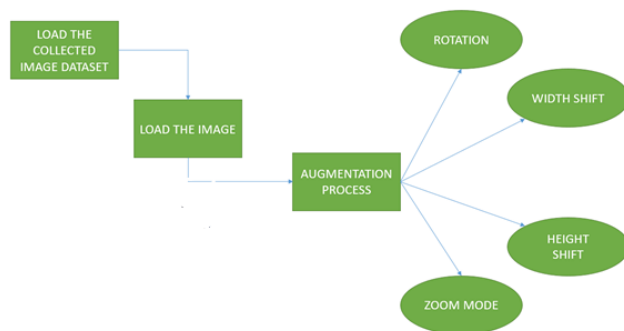


Figure 6.3 Data augmentation

Face Detection Module:

The "Ultra-Light-Fast-Generic-Face-Detector" is intended for general-purpose face detection in low-power computing devices, including Android and iOS phones and PCs (CPU and GPU). The model is a real-time universal face detection model intended for edge computing or low-power devices. It can be used for real-time familiar scene faces in low-power computing devices like ARM. Security tracking, surveillance, human-computer interaction, entertainment, and other facial recognition technology applications are standard. The first step in facial recognition is detecting human faces in digital images, and an optimal face detection model can be judged based on how easily and precisely it works.

In terms of speed, the Face-Detector-1MB stands out: the model's default FP32 accuracy (.pth) file size is 1.1MB, and the inference frame int8 is quantized to 300KB. The input resolution of 320x240 is just about 90 to 109 MFlops when it comes to model calculation. A VOC dataset produced by the WIDER FACE dataset, a face detection benchmark, was used in the Face-Detector-1MB training process. WIDER FACE comprises 32,203 images and 393,703 face bounding boxes. Pose is characterized by a high degree of scale variability and expression, occlusion, and lighting launched in 2015.

The 1MB lightweight model is available in a slim version with slightly faster simplification and a version-RFB with a modified RFB module for greater accuracy. The model was tested on

Ubuntu 16.04, Windows 10, Python 3.6, Pytorch1.2, and CUDA10.0, among other operating systems.

Characteristics

- The default FP32 precision (.pth) file size is 1.041.1MB, while the inference frame int8 is approximately 300KB.
- The input resolution of 320x240 is about 90109 M-Flops in the model's calculation.
- The model is available in two versions: version-slim (slightly faster simplification) and version-RFB (higher accuracy due to the updated RFB module).
- Pre-training models are provided that use greater face training at 320x240 and 640x480 input resolutions to better work in various environments.
- On xx export is supported, and it's simple to transplant.

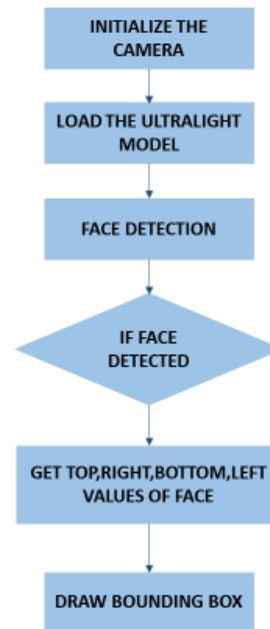


Figure 6.4 Ultralight face detector flow diagram

Face Recognition Module:

Mobile Facenet, which is more precise in classifying faces, was used to identify the face. MobileFaceNet is a neural network that achieves 99.28 percent accuracy on the labelled faces in the wild (LFW) dataset and 93.05 percent accuracy on the AgeDB dataset. On a Qualcomm Snapdragon processor, the network used around a million parameters and took just 24 milliseconds to run and generate results. Compared to ShuffleNet, which has many more parameters and takes a little longer to execute on the CPU, this achievement is 98.70 percent and 89.27 percent ShuffleNet has many options and takes a little longer to set up a run on the CPU, has a higher percentage.

The researchers made it simple to replace CNN's global average pooling layer with a depthwise convolution layer, which increases facial recognition performance. This advancement is critical as the artificial intelligence world looks for effective models that can run on the low compute power found in today's mobile phones. Compressing pre-trained networks through knowledge distillation is another method for obtaining lightweight facial verification models. On LFW with a

model size of 4.0 MB, such methods reached 97.32 per cent facial verification accuracy. MobileFaceNets achieves similar accuracy with a minimal budget, which is a remarkable achievement.

The MobileNetV2 architecture is a source of inspiration for the MobileFaceNet architecture. Our main building blocks are the residual bottlenecks proposed in MobileNetV2. The non-linearity used by the researchers is PReLU, which is better suited for facial verification than ReLU. In the beginning, the researchers often use a fast down-sampling strategy. As the feature output layer and a linear 11 convolution layer following a global linear depth wise convolution layer.

The following table shows the detailed architecture:

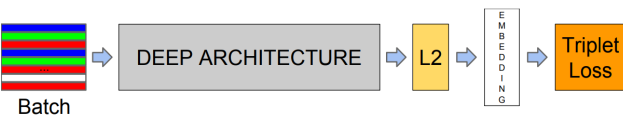
Input	Operator	t	c	n	s
$112^2 \times 3$	conv3x3	-	64	1	2
$56^2 \times 64$	depthwise conv3x3	-	64	1	1
$56^2 \times 64$	bottleneck	2	64	5	2
$28^2 \times 64$	bottleneck	4	128	1	2
$14^2 \times 128$	bottleneck	2	128	6	1
$14^2 \times 128$	bottleneck	4	128	1	2
$7^2 \times 128$	bottleneck	2	128	2	1
$7^2 \times 128$	conv1x1	-	512	1	1
$7^2 \times 512$	linear GDConv7x7	-	512	1	1
$1^2 \times 512$	linear conv1x1	-	128	1	1

Table 6.1: MobileFaceNet architecture

0.99 million parameters are used in the leading MobileFaceNet network. The researchers decided to change the input resolution from 112112 to 11296 or 9696 to reduce computational costs. MobileFaceNet also lost the linear 11 convolution layer after the linear GDConv layer. The result is a network known as MobileFaceNet M.

The baseline models used by the researchers were MobileNetV1, ShuffleNet, and MobileNetV2. For a fair performance comparison, all MobileFaceNet models and baseline models are trained from scratch on the CASIA-Web face dataset by ArcFace loss. At 60K iterations, the training is complete.

MobileFaceNet, MobileFaceNet (11296), and MobileFaceNet (9696) are also trained on the MSCeleb-1M database's cleaned training set, which contains 3.8 million images from 85,000 subjects, in order to achieve even better results. On LFW and AgeDB-30, the accuracy of our main MobileFaceNet is increased to 99.55 percent and 96.07 percent, respectively.



6.5 MobileFaceNet Architecture diagram

EMAIL INTEGRATION :

We will use the SMTP protocol to integrate Email, which is used for sending and receiving mail as configured. The following diagram depicts the workflow of that module.

The Intel Processor is a small computer that is primarily used for educational purposes. In February 2016, the company released the Intel Processor-3 model, including built-in WIFI, Bluetooth, and USB boot capabilities. Manufacturers and electronics enthusiasts quickly adopt it for projects due to its small size and low cost. Although the Intel Processor is slower

than a laptop or desktop computer, it is still a fully functional Linux machine. The Raspbian operating system is usually pre-installed on Intel processors.

Intel Processor is a powerful tool in the IoT (Internet of things) vision. Several IoT projects have been developed using Intel processors. It can also be used with IFTTT, ThingSpeak, Artik Cloud, Firebase, and Particle, among other IoT cloud platforms. The use of SMTP (Simple mail transfer protocol) for sending and receiving emails is one of the raspberry-pi/intel-applications. processor's

An email will be sent using a Raspberry Pi/Intel Processor and an SMTP server in this project. MTA (Mail Transfer Agent) and MDA (Mail Delivery Agent) provide domain searching and local delivery services, while SMTP starts a session between the user and the server.

On a TCP/IP network, SMTP is the standard protocol for delivering email services. This server allows you to send and receive email messages.

SMTP is an application-layer protocol that allows you to send and receive Email over the Internet. The Internet Engineering Task Force is in charge of keeping it up to date (IETF). SMTP is made up of four key components that are usually found in an email client application:

1. Local user or client-end utility known as the mail user agent (MUA)

2. Server known as mail submission agent (MSA) 3. Mail transfer agent (MTA)

4. Mail delivery agent (MDA)

Using an Intel Processor Python programme to receive email alerts or data set is a beneficial application. The smtplib library in the python script is all that is required. Python comes in various versions, but the 3.2 and 2.7 versions are the most compatible with the Raspberry Pi. The steps for sending SMTP email using pi are listed below:

How to Send Email with an Intel Processor

Step 1: Connect the power and LAN cables to the Intel Processor, then make a WIFI hotspot and connect to it.

Step 2: - Then, on the Pi, open the terminal window. Then, input the hostname or IP address into the putty software.

Step 3: The Intel Processor must be updated. So, run the command below to instal the most recent packages.

Step 4: - Then run the command `echo "hello" | mail -s "test" xyz@gmail.com`.

This command specifies the content of our mail and the subject and the mail id to which it will be sent.

Step 5: - Next, we'll need to make a new file in Python, which we can do with the command `nano newmailing.py`.

A different way to complete the same task

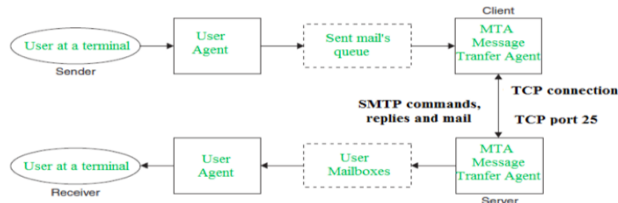
Create a new file in the Python IDE 2.7 or higher, and save it as `newmailing.py` by pressing `Ctrl + x`.

The name provided by the user when saving the file is `newmailing.py` in this case.

Step 6: Allowing Gmail SMTP Access for Standard Authentication Accounts

To grant your app access to Gmail's SMTP server.

Step 7: Log into your Gmail account and check your inbox; if everything is in order, Your email address will receive a message. The diagram below depicts the overall flow of the work.



Mobile app development:

For this project, we used React Native to create a mobile application. The javascript framework reacts native is used to create a native mobile app as an interface. The user can use the mobile app to look for available parking slots in their preferred location. It also provides the consumer with the option of creating a reserved parking spot near their location. React Native (also referred to as RN) is a popular JavaScript-based mobile app framework that allows you to create natively rendered iOS and Android apps. Using the framework, you can build applications for a variety of platforms with the same codebase.

Facebook first released React Native as an open-source project in 2015. It became one of the popular mobile development solutions in just a few years. Some of the world's most popular mobile apps, such as Instagram, Facebook, and Skype, use React Native growth. In this post, we go over these and other React Native-powered app examples in greater detail. The worldwide success of React Native can be attributed to some factors. First, companies can use React Native to generate code once and use it to power both their iOS and Android apps. This results in significant time and resource savings.

Second, React NativeNative was built on top of React, a popular JavaScript library when the mobile framework was launched. In this section, we go over the differences between React and React NativeNative in greater depth.

Finally, the framework enabled frontend developers who could previously only work with web-based technologies to create robust, production-ready mobile apps. React NativeNative is written in a hybrid of JavaScript and JXL, an XML mark-up language. The framework can interact with threads in both worlds: JavaScript-based threads and native app threads.

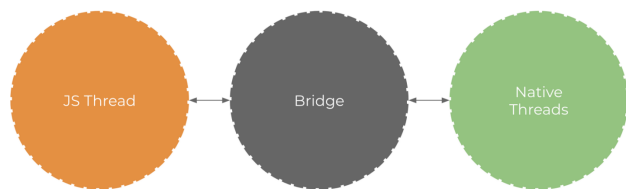


Figure 6.8: React native bridge

React Native uses a so-called "bridge". Even though JavaScript and Native threads are written in entirely different languages, bidirectional communication is possible thanks to the bridge feature.

VII. SOFTWARE DESCRIPTION

VISUAL STUDIO

As an IDE, Microsoft Visual Studio is used in this project. Visual Studio Code incorporates the ease of use of a source code editor with advanced developer functionality such as

IntelliSense code completion and debugging. It is, first and foremost, an editor who stays out of your way. Less time fiddling with your environment and more time executing your ideas thanks to the delightfully frictionless edit-build-debug cycle. Visual Studio Code is available for macOS, Linux, and Windows so that you can get started right away. Visual Studio Code's core feature is a lightning-fast source code editor that's ideal for day-to-day use. Syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more are all available in VS Code, which supports hundreds of languages. You can navigate your code with ease thanks to intuitive keyboard shortcuts, simple customization, and community-contributed keyboard shortcut mappings. When it comes to serious coding, you'll also benefit from resources that understand more code than just blocks of text. IntelliSense code completion, rich semantic code comprehension and navigation, and code refactoring are all integrated into Visual Studio Code. When the coding becomes difficult, the challenging turn to debugging, we made it happen because debugging is always the one element that developers lack the most in a leaner coding experience. You can walk through source code, inspect variables, display call stacks, and run commands in the console with Visual Studio Code's interactive debugger.

VS Code also integrates with create and scripting software to help you complete essential tasks faster. With Git support in VS Code, you can deal with source control without leaving the editor, including displaying pending changes. Customize every feature to your taste and add as many third-party plugins as you like. Although most scenarios work right out of the box, VS Code develops with you, and we allow you to customize your experience to fit your particular needs. VS Code comes with enhanced built-in support for Node.js development, including JavaScript and TypeScript, and is powered by the same underlying technologies like Visual Studio. Web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON are all well-represented in VS Code.

Visual Studio Code incorporates the best of the Internet, NativeNative, and language-specific technology in its architecture. VS Code uses Electron to combine web technologies like JavaScript and Node.js with native app speed and versatility. VS Code makes use of a newer, faster version of the same industrial-strength HTML-based editor that powers the "Monaco" cloud editor, Internet Explorer's F12 Tools, and a variety of other projects. VS Code also uses a tools service architecture that integrates with many of the same technologies that power Visual Studio, such as Roslyn for.NET, TypeScript, the Visual Studio debugging engine, and others. A public extensibility model in Visual Studio Code allows developers to create and use extensions and customize their edit-build-debug experience.

Python:

This project's programming language is Python. Python is a high-level object-oriented programming language with built-in dynamic semantics mainly used for web and app creation. It has much appeal in Rapid Application Growth because it allows for dynamic typing and binding. Python is a simple language to learn since it uses a special syntax that emphasizes readability. Python programming is a lot easier to read and understand. Translate than code written in other languages. As a result, software maintenance and implementation costs are reduced because teams can collaborate without primary language and skill barriers.

Python also encourages modules and bundles, enabling programmes to be constructed in a modular manner and code to be reused through several projects. One of Python's most attractive features is that both the standard library and the interpreter are freely accessible in binary and source form. Python and all of the required resources are accessible on all major platforms, so there is no exclusivity. As a result, it's an appealing choice for developers who don't want to be concerned with high construction costs.

Android Studio:

Android Studio is used to build mobile applications in this project. Based on JetBrains IntelliJ IDEA software and developed specifically for Android production, Android Studio is the official integrated development environment (IDE) for Google's Android operating system. In 2020, it will be available for download on Windows, macOS, and Linux operating systems and as a subscription-based service. It takes the Eclipse Android Development Tools (E-ADT) as the primary IDE for developing native Android apps. Any environment or code editor that will make writing code for your Android application easy. Android Studio is an Integrated Development Environment (IDE) that allows you to create Android apps. It's based on the IntelliJ IDEA software. Android Studio has a variety of features to help you be more productive when developing Android apps. These include:

- Instant runs on a mobile device or any virtual device to test the app's functionality.
- A wide range of review methods can improve testing before the product is released on the Play Store.
- Develop a framework based on Gradle that is flexible.
- Auto-completion of code.

Material Design Elements have been revised in the latest project models. Integrated simulator window: Instead of running in a separate window, the emulator can be run directly in the IDE. Navigation of the Dagger/Hilt code: To learn more about the Dagger and Hilt styles in your code, click on the new gutter behaviour. The studio can create code for a TensorFlow Lite model imported, making it easier to communicate with that model from the app code.

REFERENCES

1. Jingxiao Zheng, Rajeev Ranjan, Ching-Hui Chen, Jun-Cheng Chen, Carlos D. Castillo, and Rama Chellappa, "An Automatic System for Unconstrained Video-Based Face Recognition", VOL. 2, NO. 3, JULY 2020
2. Xiaojuan Cheng, Jiwen Lu, Senior Member, IEEE, Bo Yuan, Member, IEEE, and Jie Zhou, Senior Member, IEEE "Face Segmenter-Enhanced Deep Feature Learning for Face Recognition" 2019 IEEE Transaction.
3. Lei Zhang, Senior Member, IEEE, Ji Liu, Bob Zhang, Member, IEEE, David Zhang, Fellow, IEEE, Ce Zhu, Fellow, IEEE "Deep Cascade Model based Face Recognition: When Deep-layered Learning Meets Small Data" IEEE transactions on image processing, VOL. X, NO. X, AUG 2019
4. Yuqi Zhang, Yongzhen Huang, Senior Member, IEEE, Shiqi Yu, and Liang Wang, Fellow, IEEE "Cross-view Gait Recognition by Discriminative Feature Learning". Citation information: DOI 10.1109/TIP.2019.2926208, IEEE Transactions on Image Processing.

5. Christian Galea, Student Member, IEEE, and Reuben A. Farrugia, Member, IEEE "Matching Software-Generated Sketches to Face Photos with a Very Deep CNN, Morphed Faces, and Transfer Learning" IEEE transactions on information forensics and security, 2017.
6. Wasserstein CNN: Learning Invariant Features for NIR-VIS Face Recognition.
7. Jiaojiao Zhao, Jungong Han, and Ling Shao, Senior Member IEEE, "Unconstrained Face Recognition Using A Set-to-Set Distance Measure on Deep Learned Features" Citation information: DOI 10.1109/TCSVT.2017.2710120, IEEE Transactions on Circuits and Systems for Video Technology IEEE transactions on circuits and systems for video technology.
8. Jiwen Lu, Gang Wang, Weihong Deng, and Jie Zhou, "Simultaneous Feature and Dictionary Learning for Image Set Based Face Recognition" Citation information: DOI 10.1109/TIP.2017.2713940, IEEE Transactions on Image Processing.
9. Gaurav Goswami, Student Member, IEEE, Mayank Vatsa, Senior Member, IEEE, and Richa Singh, Senior Member, IEEE, "Face Verification via Learned Representation on Feature-Rich Video Frames" Citation information: DOI 10.1109/TIFS.2017.2668221, IEEE Transactions on Information Forensics and Security.
10. Jiwen Lu, Senior Member, IEEE, Junlin Hu, and Yap-Peng Tan, Senior Member, IEEE "Discriminative Deep Metric Learning for Face and Kinship Verification" Citation information: DOI 10.1109/TIP.2017.2717505, IEEE Transactions on Image Processing
11. Tong Zhang, Wenming Zheng, Member, IEEE, Zhen Cui, Yuan Zong, Jingwei Yan and Keyu Yan, "A Deep Neural Network Driven Feature Learning Method for Multi-view Facial Expression Recognition" Citation information: DOI 10.1109/TMM.2016.2598092, IEEE Transactions on Multimedia
12. Thibaud Senechal, Member, IEEE, Vincent Rapp, Member, IEEE, Hanan Salam, Renaud Seguier, Kevin Bailly, and Lionel Prevost, "Facial Action Recognition Combining Heterogeneous Features via Multi Kernel Learning", VOL. 42, NO. 4, AUGUST 2016
13. Zhen Lei, Member, IEEE, Dong Yi and Stan Z. Li, Fellow, IEEE, "Learning Stacked Image Descriptor for Face Recognition" [Vol.no: 2473415, 2015]
14. Weitao Xu, Member, IEEE, Yiran Shen*, Member, IEEE, Neil Bergmann, Senior Member, IEEE, and Wen Hu, Senior Member, IEEE, "Sensor-assisted Multi-view Face Recognition System on Smart Glass" [Vol.no: 1536-1233, 2015]
15. Javier Galbally, Sébastien Marcel, Member, IEEE, and Julian Fierrez "Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition" [Vol.no 1057-7149, 2015].

