Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 04, April 2021: 2200-2218

Research Article

Unique Attendance System with Group Marking of the Students Presence With a Time Saving Approach

L.Raji[1], Rahul Maddula[2], Pinupolu Ramanaiah Saiteja[3], Akshay Kumar T N[4]

[1]Assistant Professor department of Computer Science and Engineering, R.M.K. Engineering College Kavaraipettai, India
[2,3,4]fourth year, Department of Computer Science and Engineering

Abstract

Monitoring the patient remotely, has always been a challenging issue in the health care domain. Especially, fall detection during monitoring, is a vital aspect. Progress in technology, has facilitated vision-based systems, and thereby detecting a fall and providing a timely help to the concerned person is possible. This paper mainly focusses on the implementation of a computer vision-based system, which also incorporates Machine Learning algorithms to learn the actions of a person, thereby detecting a fall, and notifying the primary caregiver during the situation, right away, so as to provide the help at the right time. The implemented system, is able to detect the actions of a user by classifying the broad range of activities into fall and non-fall actions.

Keywords: *Fall Detection, Machine Learning, Pose estimation*

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 developing an attendance management scheme is to bring the conventional way of taking attendance into the twenty-first century. Another justification for developing this software was to have the report produced automatically at the end of the session or in the middle of it.

The system being developed is cost-effective from the standpoint of a school or college.In the way that all bureaucracy has been removed, it is cost-effective. Since the measurements are automatic and completed at the end of the month or if required by the customer, the device saves time. As the data is required, the obtained results have few errors and are highly accurate. The method does not necessitate any additional preparation. The attendance of the entire session is recorded in a log, and reports are provided at the conclusion of the session. We don't want to generate reports in the middle of a session or in response to a requirement so it takes longer to measure. 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. To produce the report at the end of the session, we'll need further estimates. Furthermore, the pupil is not given any chance to change his or her attendance. There is a greater risk of error since all measurements to produce the report are performed manually. A considerable amount of documentation is needed under the new scheme. Since all of the papers were needed to produce the reports, even the lack of a single register/record caused a challenge. We can't produce reports in the middle of a session or when required because all work is performed manually, which takes a long time.

Characteristics of Attendance Management System: Since data retrieval and storage are fast and data is easily stored, the proposed solution is user-friendly. Furthermore, the proposed system includes a graphical user interface, allowing users to interact with the system with ease. The proposed system reports are simple to create, and the user can produce them as required (monthly) or even in the middle of a session. The user should send an email to students encouraging them to come to class on a regular basis. The new scheme only necessitates a small amount of paperwork. All data is entered into the database as soon as possible, and reports can be produced using machines.

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.

Lıterature Revıew

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. To address these issues, facial/fiducial detection, face affiliation, and face recognition modules form a comprehensive and functional platform for unconstrained video-based face recognition. The observed faces are then clustered together using multi-shot image-specific face association approaches. Finally, the proposed face matcher understands the faces subspace to subspace similarity metric using an unsupervised subspace learning strategy and a neural network. Extensive tests on challenging video datasets, such as the Multiple Biometric Grand Challenge, Face and Ocular Challenge Series and IARPA JANUS Benchmark B for multiple-shot videos, show that the proposed system can accurately detect and associate faces from unconstrained videos and multiple-shot videos.

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.

For localised facial property, a Face Segmenter-Enhanced Network (FSENet) was created to exploit face recognition. Due to major intra-class differences and inter-class fine-grain, most emerging approaches stress the holistic characteristics of whole face pictures, restricting discriminative ability. To counter this, we present a face section that can be used to parse the face into local components and investigate their internal similarities, enhancing the capacity to distinguish between people. A semantic parsing module is implemented, which assigns a semantic component label to each pixel. To boost personalised characteristics, we will continue to improve the structural correlation of facial part elements. Finally, we add holistic and local data to improve the face descriptor's discriminative capacity. Extensive tests on well-known public-domain datasets, such as marked Face in the Wild (LFW), youtube Faces (YTF), IARPA IJB-A, IJB-B, and IJB-C, and the MegaFace Challenge, prove that our approach performs well.

Pros:

This method achieves promising performance. Cons:

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

Existing System

Although deep learning algorithms have surpassed humans in still image-based face recognition, unconstrained video-based face recognition remains a challenge. Due to the vast amount of data to be analysed and intra/inter-video inconsistencies in posture, illumination, occlusion, scene, blur, video quality, and other factors, this is a difficult challenge. In this post, we look at challenging conditions for video-based facial recognition in surveillance footage with low-quality frames and multiple-shot images. We propose a robust and efficient interface for unconstrained video-based face recognition that includes modules for face/fiducial detection, face affiliation, and face recognition to overcome these concerns. In order to identify faces in videos effectively, we use multi-scale single-shot face detectors.Faces are identified and grouped using facial association methods that have been carefully designed and are useful for multi-shot photographs.

3.1 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.

3.2 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.Face detection may be used to identify people in an organisation for attendance purposes. In an organization's performance assessment, maintaining and evaluating attendance records is important. The Automated Attendance 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.

3.3 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.

3.4 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.

Tehnology Used OpenCV

In 1999, Gary Brodsky created OpenCV at Intel, and the first version was released in 2000. Vadim Pisarevsky has taken over as head of Intel's Russian OpenCV tech unit from Gary Brodsky. Stanley, the winning car in the 2005 DARPA Grand Challenge, used OpenCV. It was later successfully built with Willow Garage's assistance, with Gary Brodsky and Vadim Pisarevsky at the helm. OpenCV is free for both commercial and academic use since it is released under the BSD licence. It supports Windows, Linux, Mac OS, iOS, and Android and has C++, C, Python, and Java interfaces. OpenCV was designed with real-time applications and high numerical efficiency in mind. The library can take advantage of multi-core computing since it is written in optimised C/C++. OpenCV has been widely embraced, with a user base of over 47 thousand people and over 6 million downloads estimated. Interactive sculpture, mine inspection, web map sewing, and advanced robotics are among the applications

4.1 OpenCV-Python

Guido van Rossum developed Python, a general-purpose programming language.

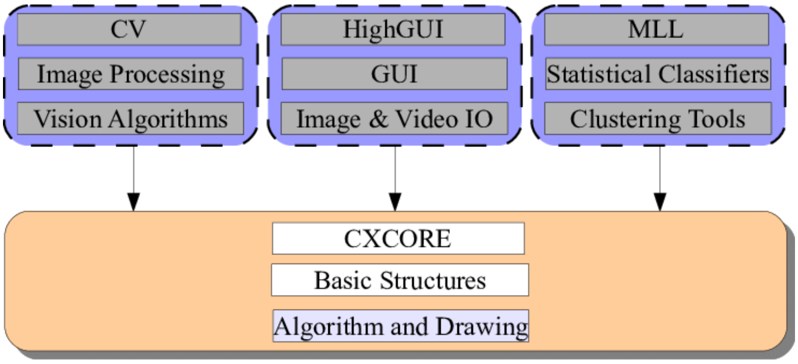


Figure 1 OpenCV software architecture

4.2 Face detection

So, whatever operations Numpy can do, you can combine them with OpenCV to expand your arsenal of weapons. Other Numpy-supporting libraries, such as SciPy and Matplotlib, can also be used. As a result, OpenCV-Python is a great way to prototype computer vision issues easily.

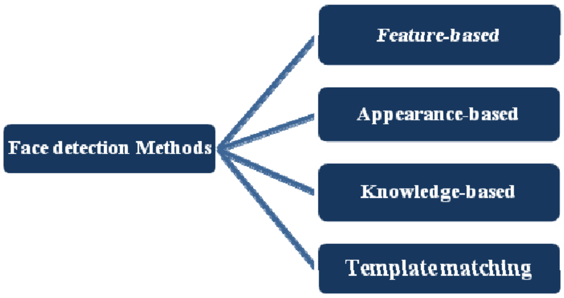


Figure 2 Face detection methods

4.3 Knowledge-Based

The knowledge-based approach is based on human experience and uses a collection of rules to detect faces. Due to its convenience and code readability, a face must have a nose, eyebrows, and mouth that are all within a given time frame. Python, on the other hand, has the advantage of being conveniently extensible for C/C++.

OpenCV-Python works in this way; it's a Python extension for the original C++ implementation. Additionally, Numpy's assistance helps the job go more smoothly. Numpy is a highly optimised numerical operations library. It produces syntax in the MATLAB format. OpenCV converts all array structures to and from Numpy arrays. So, whatever operations Numpy is capable of, you can combine them with OpenCV to expand your arsenal of weapons. Aside from that, many other Numpy-supporting libraries, such as SciPy and Matplotlib, can be used for this. As a result, OpenCV-Python is an excellent platform for prototyping computer vision issues quickly. The challenge in constructing a satisfactory set of rules is the biggest drawback of these approaches. There might be a lot of false positives if the rules were too vague or too narrow. For identifying multiple faces in multiple photographs, this approach is inadequate.

4.4 Feature-Based

The feature-based approach locates faces by removing structural features from the face. It is first qualified as a classifier before being used to differentiate between facial and non-facial regions.

The aim is to break free from the tie. Face recognition's inherent limitations. The success rate of this process, which was broken down into various stages and included images of several faces, was 94 percent.

4.5 Template Matching

Using pre-defined or parameterized face models, the Template Matching system locates or identifies faces by correlating the templates and input photos. The four parts of a human face, for example, are the pupils, facial contour, nose, and mouth. The edge detection tool can also be used to build a face model completely out of edges. While this method is simple to use, it falls short when it comes to facial recognitionOn the other hand, deformable structures have been proposed as a solution to these issues.

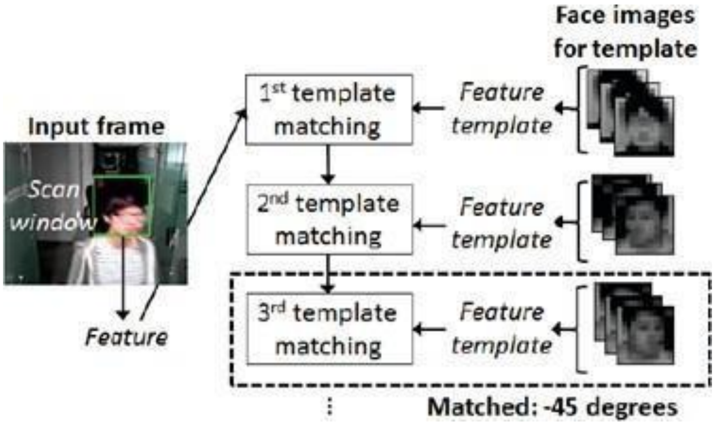


Figure 3. Template Matching

4.6 Appearance-Based

To find face models, the appearance-based approach uses a set of delegate preparation facial images. Other evaluation approaches are outperformed by the appearance-based methodology. To assess the related characteristics of face pictures, appearance-based approaches depend on statistical analysis and machine learning techniques. This approach is also used in facial recognition to extract features.

4.7 Face recognition

Face recognition and face analysis are often confused issues. Face Recognition, on the other hand, uses a database of faces to verify the input face and decide whether it is recognised or unknown.

4.8 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). Many different algorithms were created as researchers' interest in face recognition increased, three of which have been extensively studied in the face recognition literature.

There are two approaches to recognition algorithms:

1. Geometric : This is focused on the spatial arrangement of facial features or the geometrical relationship of facial landmarks. The eyebrows, nose, and mouth, which are the most prominent geometrical features of the face, are seen first. Faces are first identified, then categorised based on geometrical distances and angles between features.

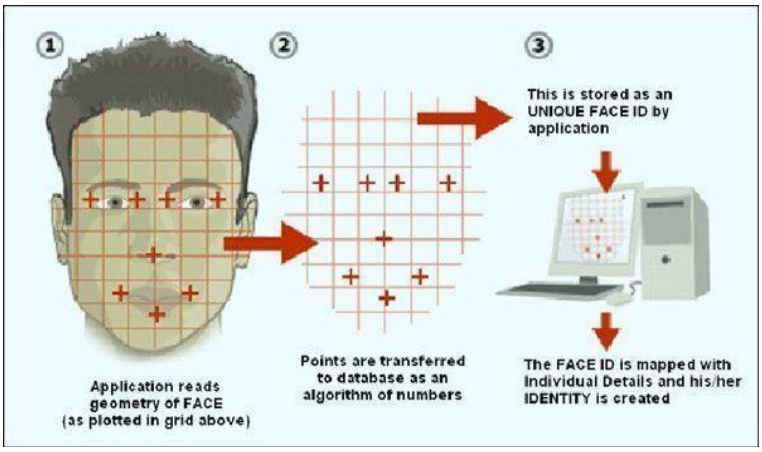

1
2
3
This is stored as an UNIQUE FACE ID by application
Application reads geometry of FACE (as plotted in grid above)
Points are transferred to database as an algorithm of numbers
The FACE ID is mapped with Individual Details and his/her IDENTITY is created

Figure 4 Geometric face recognition

1. 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.

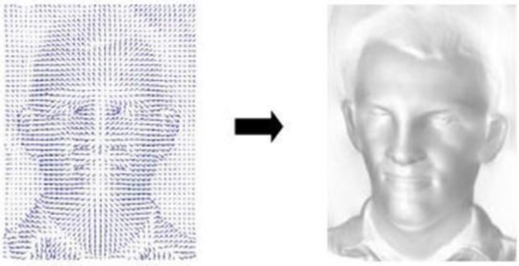


Figure 5 Photometric stereo for face recognition

4.9 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.

System Design

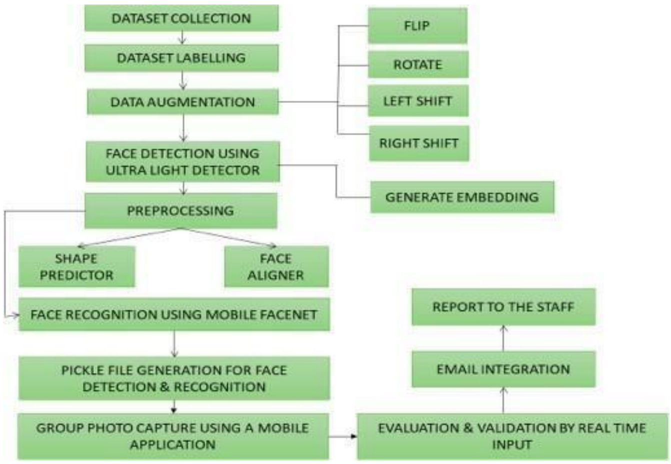

DATASET COLLECTION
DATASET LABELLING
DATA AUGMENTATION
FLIP
ROTATE
LEFT SHIFT
RIGHT SHIFT
FACE DETECTION USING ULTRA LIGHT DETECTOR
GENERATE EMBEDDING
PREPROCESSING
SHAPE PREDICTOR
FACE ALIGNER
FACE RECOGNITION USING MOBILE FACENET
PICKLE FILE GENERATION FOR FACE DETECTION & RECOGNITION
GROUP PHOTO CAPTURE USING A MOBILE APPLICATION
EVALUATION & VALIDATION BY REAL TIME INPUT
EMAIL INTEGRATION
REPORT TO THE STAFF

Figure 6 Architecture Diagram

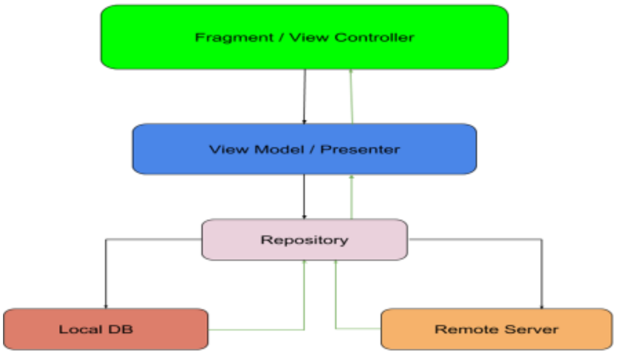


Figure 7 React native Architecture Diagram

4.10 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.

4.11 Modules Description:

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

4.12 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. With enough experience, a deep network will segment and recognise the "key points" of any individual in a picture. Our model performs better when there is more labelled data available. 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. Another alternative is to use an image annotation GUI that already exists.

4.13 Data Augmentation:

As more data becomes available, the performance of deep learning neural networks increases. A strategy for extracting new training data from old data is known as data augmentation. This is accomplished by using domain-specific methods to translate examples from the training data into new and special training examples.

The most well-known method of data augmentation is image data augmentation, which entails transforming images from the training dataset into transformed copies that are labelled similarly to the original image.

Transforms provide operations such as shifts, twists, zooms, and other image processing operations. Since the model is impossible to see an upside-down cat shot, a sideways flip of a cat photo makes no sense and is almost definitely unacceptable.Convolutional neural networks(CNN), for example, are modern deep learning algorithms, Can learn independent features of where they appear in the picture.

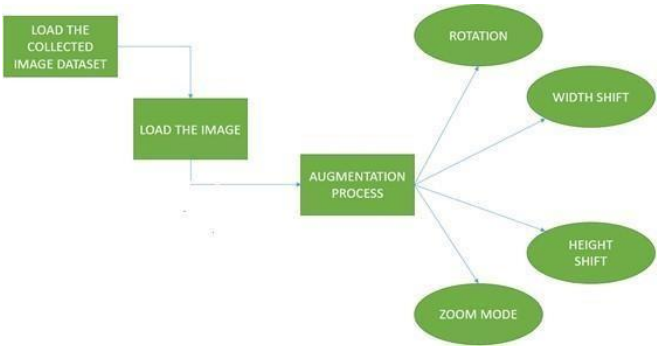


Figure 8 Data augmentation

4.14 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 to recognise human faces in digital images, and an ideal face detection paradigm can be judged on how well it performs.

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.

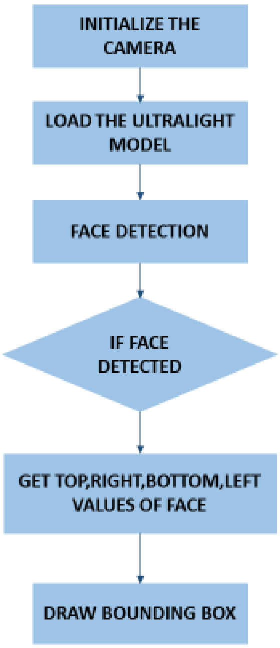


Figure 9 Ultralight face detector flow diagram

4.15 Face Recognition Module:

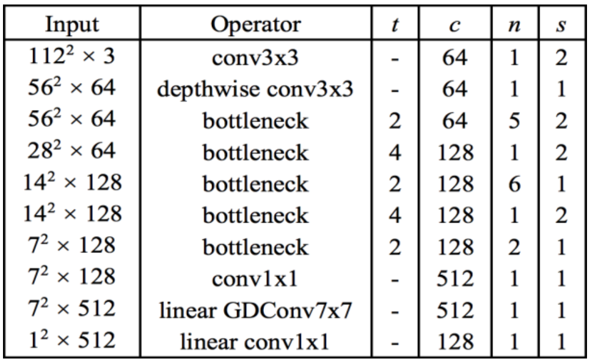
Mobile Facenet, which is more precise in classifying faces, was used to identify the face. MobileFaceNet is a neural network that achieves 99.28 percent accuracy on the labelled faces in the wild (LFW) dataset and 93.05 percent accuracy on the AgeDB dataset. On a Qualcomm Snapdragon processor, the network 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:

Table 10. MobileFaceNet 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

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 Conv 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.

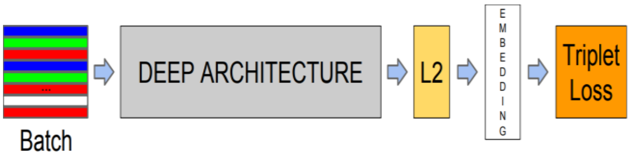


Figure 10. MobileFaceNet Architecture diagram

4.16 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. 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. An email will be sent using a Raspberry Pi/Intel Processor and an SMTP server in this project.SMTP is the basic protocol for providing email services over a TCP/IP network. You may use this server to send and receive email messages.

SMTP stands for Simple Mail Transfer Protocol, and it helps 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

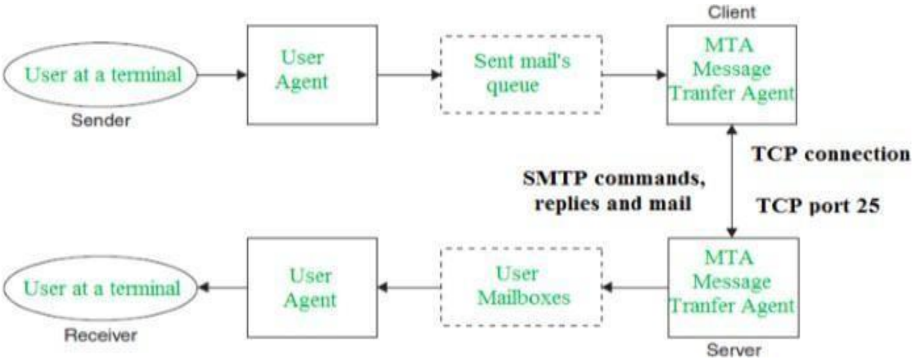
Create a new file in the Python IDE 2.7 or higher, and save it as newmailing.py by pressing Ctrl + x.

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.



4.17 Mobile app development:

For this project, we used React Native to create a mobile application. The javascript framework react 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 was first released as an open-source project by Facebook in 2015. It became one of the popular mobile development solutions in just a few years. It is used by the world's most common smartphone applications, including Instagram, Facebook, and Skype. The framework can interact with threads in both worlds: JavaScript-based threads and native app threads.

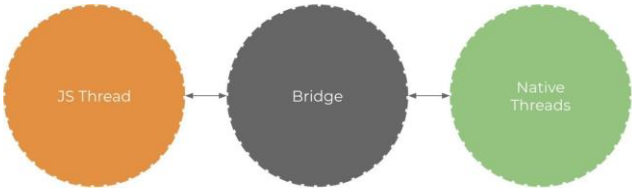


Figure 12 React native bridge

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

Software Description Visual Studio

Microsoft Visual Studio is included in this project as an IDE.

Visual Studio Programming integrates the ease of use of a source code editor with advanced development tools such as IntelliSense code completion and debugging. You'll spend less time fiddling with your environment and more time putting your thoughts into action thanks to the pleasantly frictionless edit-build-debug loop.

Visual Studio Code, which is available for macOS, Linux, and Windows, allows you to get started right away. Visual Studio Code's main feature is a lightning-fast source code editor that's ideal for daily use. Syntax highlighting, bracket matching, auto-indentation, box selection, snippets, and more are all present in VS Code, which supports hundreds of languages. Thanks to intuitive keyboard shortcuts, basic customization, and community-contributed keyboard shortcut mappings, you can access the code with ease.

5.1 Python:

This project's programming language is Python. It is a high-level object-oriented programming language with built-in dynamic semantics that is mostly used to develop websites and apps. Since it allows for dynamic typing and binding, it has a lot of appeal in Rapid Application Growth. Python is an easy language to understand because it employs a readability-focused syntax. 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 interpreters 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.

5.2 Android Studio:

In this project, Android Studio is used to create mobile apps. Android Studio is Google's official integrated programming environment (IDE), built on JetBrains IntelliJ IDEA software and developed specifically for Android production.In the year 2020, it will be available as a subscription-based service for Windows, macOS, and Linux operating systems. Its main IDE for native Android app development is Eclipse Android Development Tools (E-ADT). Any environment or code editor that will make writing

code for your Android application easy. It's based on the Intelligence 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,

[10] 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.

[11] 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

[12] 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

[13] 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]