WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Smart Face Recognition Using IOT and Machine Learning

Ashish Kumar[1], Sylvester Kalumbi[2], Manish Kumar Gupta[3], Prof Panduranga RaoMV[4]

[1]Department of Computer Science & Engineering (Internet of Things), Jain Deemed to-be University, Bangalore,562112, India
[2]Department of Computer Science & Engineering (Internet of Things), Jain Deemed to-be University, Bangalore,562112, India
[3]Department of Computer Science & Engineering (Internet of Things), Jain Deemed to-be University, Bangalore,562112, India
[4]Department of Computer Science & Engineering (Internet of Things), Jain Deemed to-be University, Bangalore,562112, India
DOI: https://doi.org/10.55248/gengpi.4.523.42625

ABSTRACT

A fast-expanding subject called "Smart Face Recognition" uses IoT and machine learning to reliably identify people based on their facial traits. Security, retail, and healthcare are just a few of the sectors that are using this technology to boost customer satisfaction and boost productivity. IoT and machine learning work together to enable the collection of enormous volumes of data from numerous sources, including cameras and sensors, and utilize this data to train algorithms that can precisely identify people in real-time. Due to its precision, speed, and scalability—essential characteristics for applications like security and access control—this technology is gaining popularity.

The capability of this technology to learn and adapt over time is one of its main advantages. The algorithms improve in accuracy and can better recognise people even in difficult situations, including poor light or partial obstruction, as more data is gathered and analyzed.

Smart facial recognition technology has the ability to transform a number of industries and improve the user experience by delivering individualized services, safe and effective access management, and real-time monitoring. We can anticipate seeing this technology used in many brand-new and fascinating applications as it Develops.

Key Word: OpenCV, Face Recognition, FTDI Programmer, Esp-32, Security, Algorithm, Sensor, Camera.

1. INTRODUCTION

1.1. Overview

A face recognition system is a piece of technology that can recognise and match human faces from a digital image that has been uploaded to a database. A subcategory of Biometric security is facial recognition.

1.2. Problem Definition

Our goal is to create a system that uses machine learning and the Internet of Things (IoT) to recognise faces in an intelligent and effective way. Traditional face recognition systems require manual input, which can be time-consuming and error-prone, and have low accuracy. We can create a system that can recognise people accurately without user input by learning from patterns in facial features thanks to the growing use of IoT devices and machine learning techniques. Applications for this system include attendance tracking, security systems, and personalized marketing. The difficulty, however, lies in creating a system that is trustworthy and secure while simultaneously protecting the privacy of users.

1.3. Objectives

Improving the security and productivity of various businesses and public spaces is the goal of deploying smart facial recognition utilizing IoT and machine learning. The technology uses cameras and sensors to take pictures and videos of people. Machine learning algorithms are then used to identify the faces in the images and videos and compare them to a database of well-known people.

Organizations can increase their security by integrating smart facial recognition to monitor and track employee attendance, detect and prevent unauthorized access, and quickly spot possible risks. The technology also makes it possible to control access quickly and easily to places that are prohibited, such banks, airports, and government facilities.

Additionally, IoT-enabled smart facial recognition can improve the consumer experience across a range of sectors, including hospitality, retail, and healthcare. Based on the faces and preferences of the clients, it can be utilized to customize services and recommendations. IoT and machine learning are being used to develop smart facial recognition with the overall goal of enhancing security, productivity, and customer experience across a range of commercial sectors and public spaces.

1.4. Hardware and Software Tools Used

Hardware Tools	Software Tools
● NodeMCU	Language used: Python 3.0 or higher
● Web Camera(ESP32S)	Editor used: PyCharm
● Processor (Laptop)	Library used: cmake, dlib 19.18, face-recognition,numpy, opencv-python
● LED Light (Load)	Arduino ide
● FTDI Programmer	

Fig.-1.1

II.LITERATURE SURVEY

2.1. Related Works

1. **Convolutional Neural Networks for Face Recognition:**

The accuracy of face recognition systems has recently been increased by the use of convolutional neural networks (CNNs). In a study by Taigman et al. (2014), a deep convolutional neural network was trained on a sizable dataset of face photos to reach state-of-the-art performance in face identification and verification tasks. According to the study, the deep neural network was able to pick up highly discriminative elements from facial photos, which led to better performance.

2. **IoT-based Facial Recognition Systems:**

In recent years, facial recognition systems capabilities have been improved with the help of the Internet of Things (IoT). In a study by Fan et al. (2018), Raspberry Pi was utilized to construct an IoT-based facial recognition system that was used to record and process facial photographs.

The technology was able to identify faces quickly and accurately in real-time.

3. **Facial Recognition using GANs:**

The accuracy of facial recognition systems has recently been increased with the use of generative adversarial networks (GANs). Zhang et al. (2018) developed a GAN-based method to create high-quality facial images, which were subsequently used to train a facial recognition system.

The study demonstrated that the application of GANs enhanced facial recognition task accuracy.

4. **Hybrid Face Recognition Systems:**

Additionally, hybrid facial recognition systems that use both conventional computer vision methods and machine learning algorithms have been developed. In a study by Zhou et al. (2018), Local Binary Patterns (LBP) and Support Vector Machines (SVMs) were combined to create a hybrid face recognition system. In facial recognition tasks, the system was able to attain excellent accuracy.

2.2. Existing System

Five unique algorithms have been selected based on the most prevalent criteria. The approaches include Wavelet, Linear Discriminant Analysis (LDA), Principal Component Analysis (PCA), and Artificial Neural Networks (ANN). Several other criteria have been used to evaluate the algorithms. The specifications include kind and size.

A sophisticated facial recognition system camera takes a picture of a person's face when they come close to it. The edge gadget then processes the image, extracting face traits like the position of the mouth, nose, and eyes. After that, the cloud server receives this data for analysis.

The cloud server contrasts the photographs saved in the database with the facial traits that were extracted. In the event that a match is made, the system identifies the person and offers any extra data that is kept in the database. If no match is discovered, the system may ask the user for more details or it may just deny the request.

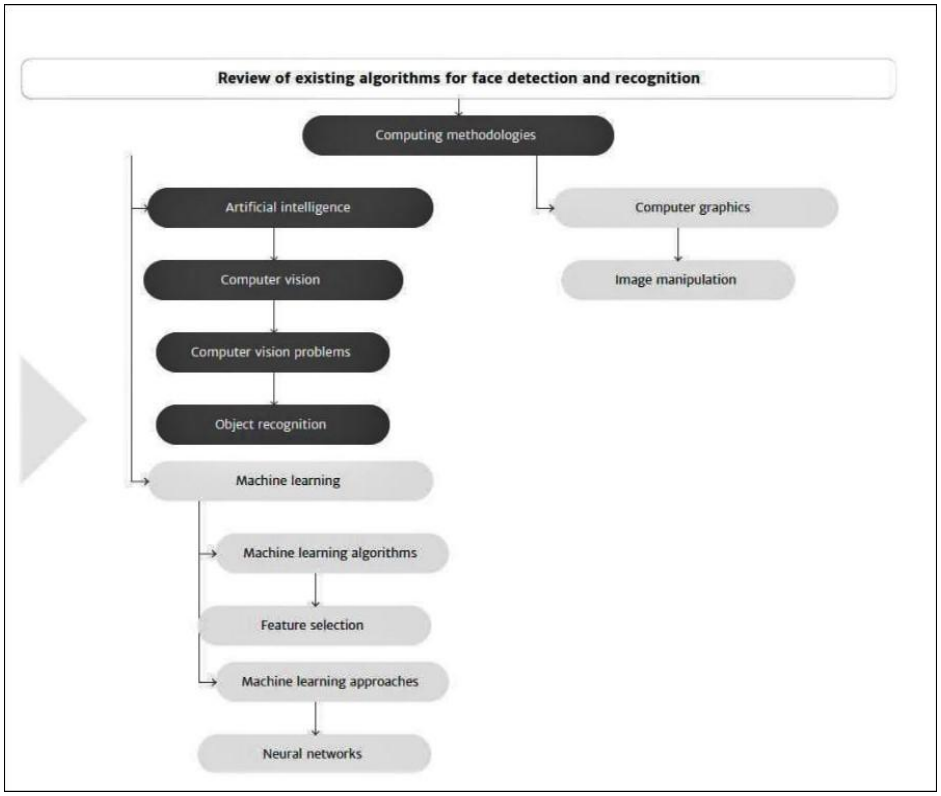

Fig.-2.1

2.3. Limitation of Existing System

- Dependence on High-Quality Data: One drawback of current IoT-based and machine learning-based smart face recognition systems is that in order to perform effectively, they need high-quality data. The algorithm might not be able to recognise the face effectively if the input data is of poor quality or is not well-structured, which could result in incorrect identifications or rejections.
- Sensitivity to environmental factors and illumination: The performance of smart facial recognition systems is greatly influenced by the lighting and other aspects of the environment.. The accuracy of the system can be considerably impacted by elements such as shadows, glare, and variations in illumination. Inappropriate outcomes may emerge from the system's failure to operate at its best in specific settings or under specific lighting conditions.
- Limited Robustness to Face Variations: The accuracy of face recognition systems can be increased by using machine learning methods, but their robustness to major variations in the faces they can recognise is constrained. Variations like facial hair, haircut changes, and facial expressions might present problems for the system and cause recognition mistakes.
- Privacy Issues: The possibility for privacy infractions is a major drawback of clever facial recognition systems. These technologies might record face information without the user's knowledge or agreement, which could result in data misuse. Furthermore, there is a chance that the system could be breached, resulting in the misuse of facial Data.
- Ethical and Legal Implications: The employment of smart facial recognition systems presents ethical and legal questions, particularly in light of problems with bias, discrimination, and spying. Depending on their ethnicity, gender, or other traits, users of the system might be treated unfairly. The use of the technology for surveillance purposes may also violate people's right to privacy, which could have negative legal and ethical repercussions.

2.4. Proposed System

1. **Data Acquisition:**

The first step in the face recognition system is data acquisition. The system must capture imagesof faces from different angles and in various lighting conditions. The data acquisition can be done using a camera or a video camera.

2. **Face Detection:**

In the second step, the system detects the face in the acquired image. Face detection algorithms use a variety of methods, such as Viola-Jones algorithm, deep learning-based algorithms like CNN, YOLO, and SSD, etc., to detect faces in an image.

3. **Face Alignment:**

After detecting the face, the next step is to align it properly. Face alignment is crucial because it helps in eliminating variations in face positions and scales. There are various face alignment techniques like MTCNN, 3D Morphable Model, etc.

4. **Feature Extraction:**

The fourth step is to extract features from the face. Feature extraction algorithms use a variety of methods such as Local Binary Pattern (LBP), Scale-Invariant Feature Transform (SIFT), Histogram of Oriented Gradients (HOG), Convolutional Neural Networks (CNNs), etc., to extract facial features.

5. **Face Recognition:**

Once the features are extracted, the face recognition algorithm matches the extracted features with the pre-existing database of known faces to identify the person. The recognition process can be performed using various methods, such as Eigenfaces, Fisherfaces, Local Binary Patterns Histograms (LBPH), Deep Neural Networks, etc.

6. **Decision Making:**

The last step is decision making. After face recognition, the system decides whether the face matches any of the known faces in the database or not. If the face matches, the system can return the identity of the person, and if not, it can mark the face as unknown.

Overall, this block diagram outlines the general flow of a face recognition system, from data acquisition to decision making. Depending on the specific implementation, there may be variations in the algorithms and techniques used.

7. **Novel algorithm**
8. **Deep Learning Methods**
9. **Convolutional neural networks (CNN)**
10. **A new algorithm to boost effectiveness and precision**

3. METHODOLOGY

3.1. Dataset

Researchers from the fields of psychology, pattern recognition, neural networks, computer vision, and graphics have all been drawn to the difficult yet fascinating topic of face identification.

- The face recognition techniques listed below are employed.
- Oriented gradient histograms (HOG) for finding all the faces
- Putting on a show and presenting your Face Landmark Estimation
- Faces are encoded
- Obtaining the individuals name from encoding

3.2. Architecture

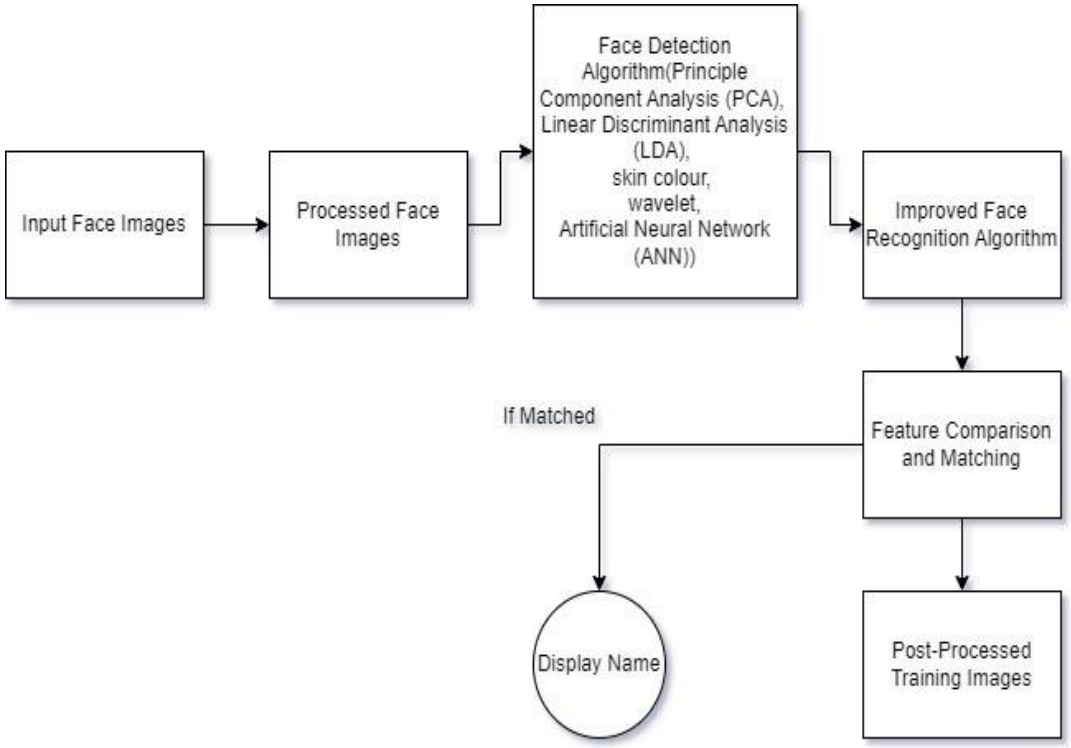

Fig.-3.2.0

3.3. Sequence Diagram

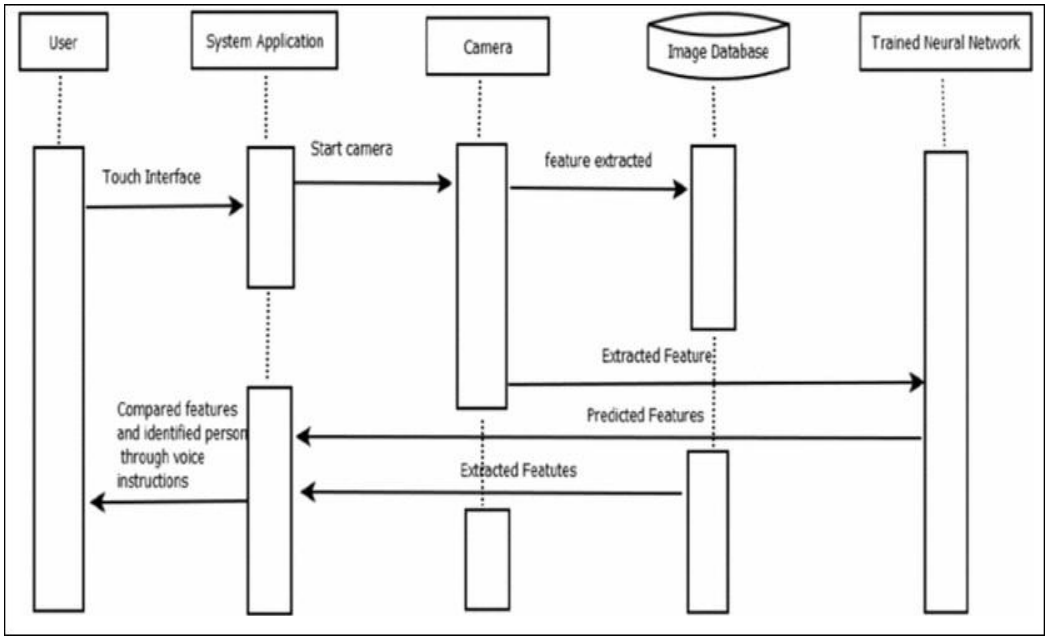

Fig.-3.2.1

Schematic Diagram Of Face Recognition:

Utilize an FTDI programmer to connect the ESP32-CAM board to your computer.

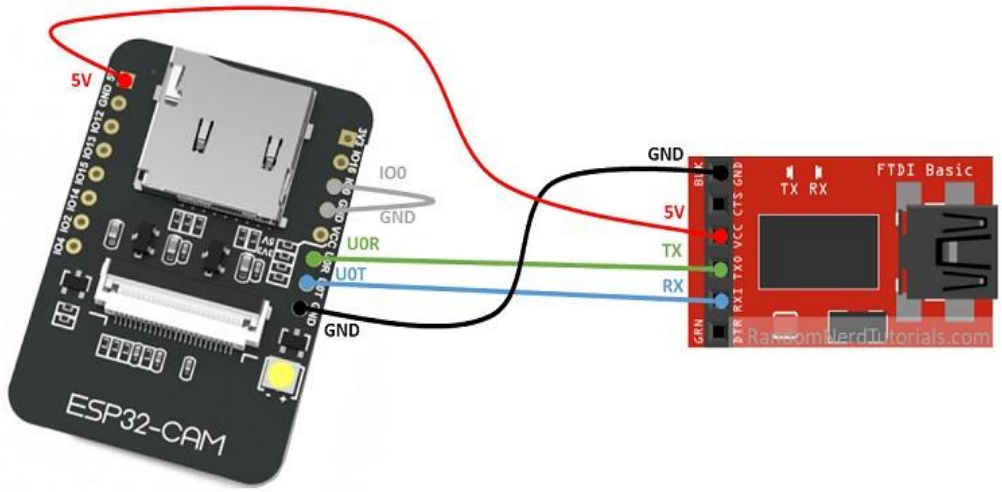

Fig.-3.2.2

A jumper that lets you choose between 3.3V and 5V is found on several FTDI programmers. Make sure the jumper is in the appropriate position to choose 5V.

4. TOOL DESCRIPTION

4.1. Hardware Requirements

- Smooth processing requires Random Access Memory (RAM) of 4 GB or more.
- A hard drive with a minimum capacity of 10 GB must be used to store all the data.
- A Dual Core Processor is required, or the most recent and advanced model should be used.
- You can use either the most recent version of MacOS or Windows (7/8/10).
- You can use a camera with at least 2 MP or more.

ESP-32:

System-on-a-chip (SoC) microcontroller ESP32 is a low-cost, low-power device that may be applied to a number of tasks, including face recognition. It is a well-liked option for IoT and smart home projects due to its built-in WiFi and Bluetooth connectivity.

The ESP32 can be utilized for facial recognition in order to take pictures with a camera module and process them with a deep learning algorithm. Real-time facial recognition and identification can be trained into the algorithm. After that, the ESP32 can carry out more face recognition-related tasks or start an action based on the profile of the individual who was Identified.

Fig.-4.1.0

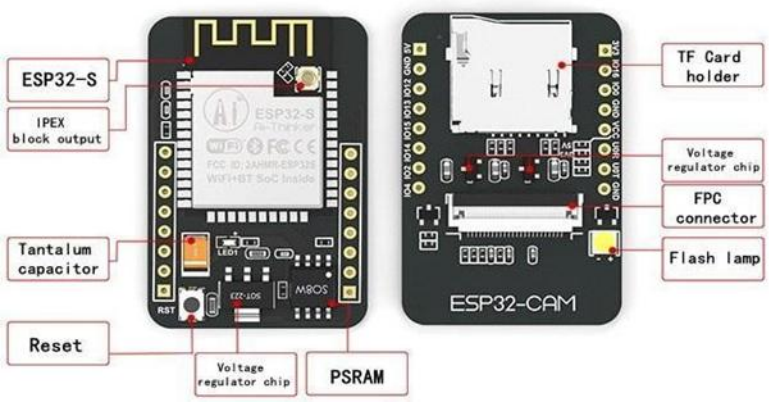

Fig.-4.1.1

FTDI Programmer:

A variety of USB-to-serial converter chips, such as those made by FTDI (Future Technology Devices International), are frequently used to offer a USB interface for microcontrollers like the ESP32.

The USB-to-serial converter chip that is built into the ESP32 development boards is referred to as FTDI when discussing the ESP32. With the aid of this chip, the ESP32 is able to connect to a computer through USB and can be updated with firmware or interfaced with via a terminal programme.

We require an FTDI programmer to upload code through the U0R and U0T pins (serial pins) of the ESP32-CAM because it lacks a USB port**.**

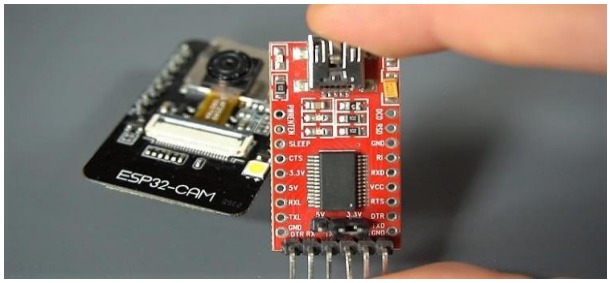

Fig.-4.1.2

4.2 Software Requirements: Description:

Arduino IDE:

A platform for open-source electronics that uses user-friendly hardware and software is called an Arduino. It is intended to make it simple for individuals to develop interactive electronic projects. A broad variety of sensors, motors, and other hardware components can be interfaced with Arduino boards microcontrollers, which can be programmed using a number of programming languages, including C++.Building projects using face recognition can be done using the Arduino microcontroller platform. Face recognition cannot, however, be done directly on an Arduino platform. It can function as a control hub to process signals from cameras and other sensors and interface with them.

You would need to attach a camera module, such as the OV7670, to the Arduino board in in order to implement facial recognition. In order to process the camera photos and run the facial recognition algorithms, you would also need to use a software library like OpenCV. After receiving the findings of the face recognition algorithm, the Arduino can take any necessary steps in accordance with those results.

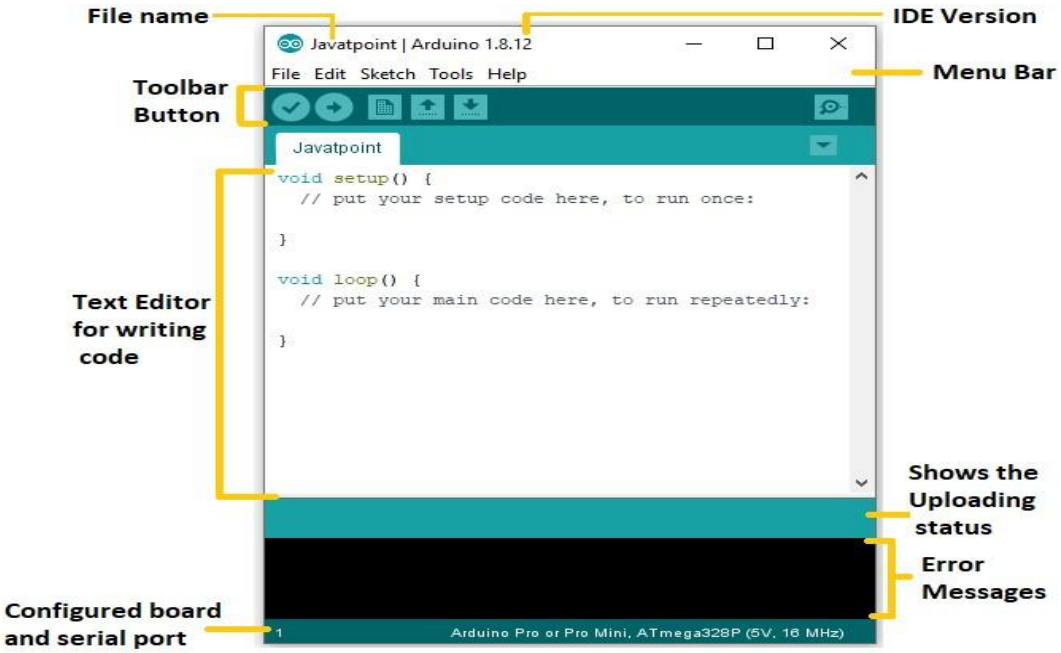

Fig.-4.2.1

Language Used:

Python is a popular programming language that can be used to develop smart face recognition systems using IoT and machine learning. Face recognition is a technique that can be used in a variety of fields, including marketing, access control, and security, to identify or confirm a person's identity based on their facial traits.

Face identification and classification using machine learning methods is still possible with face recognition systems using predefined features. To match facial traits with known individuals in a database, for instance, a face recognition system could utilize a support vector machine (SVM) classifier. The systems IoT component might have cameras or other sensors to record facial photos or data, which Python scripts could then use to process.

Editor Used:

PyCharm: Python is a computer language that may be used with PyCharm as an integrated development environment (IDE). It provides an effective editor with facilities for code completion,debugging, and intelligent code analysis.. Data preparation, face recognition machine learning models, and other software development tasks may all be carried out using PyCharm. PyCharm is also used to create software components that connect with IoT devices. Additionally, PyCharm supports many Python libraries and frameworks that are often used in machine learning and computer vision, including OpenCV and TensorFlow.

Library used

CMakeis a build system tool used to create build files for C++ programming, not a library. In order to create projects involving image recognition, especially intelligent face recognition, libraries like OpenCV, TensorFlow, and other machine learning libraries are frequently utilized in conjunction with it.It also creates native workspaces and make files that can be used in any compiler environment. In order to provide an effective, cross-platform build environment for open-source programmes like ITK and VTK, Kitwara developed the Cmake tool set.

Dlib:

In the programming language C++, the Dlib general-purpose cross-platform software library was created. Ideas from component-based software engineering and design by contract have had a significant influence on its creation. Thus, it is primarily a collection of independent software components. Open-source software is what it is.

Face-recognition: A technology known as facial recognition employs algorithms and machine learning to recognise people by their facial traits. Security systems, law enforcement, and many other applications that need for the identification and verification of people frequently employ it.

To match faces with known identities, the facial recognition system would rely on pre-programmed rules and algorithms. As a result, the system would be unable to create its own rules or models, which would prevent it from learning from new data and adjusting to it.

Instead, it would rely on a predetermined set of guidelines and information to carry out its Task.

NumPy:

Numpy is a Python library that is commonly used for numerical computing and scientific computing. It offers quick and easy array manipulation, which is highly beneficial for processing data in machine learning. Regarding the processing of image data utilizing IoT and machine learning for smart face recognition. NumPy can be used, for instance, to conduct image filtering, image reshaping, and grayscale conversion.The research and how it contributed to the outcomes both used NumPy. It could be helpful to give particular instances of how NumPy was applied in the study, such as face picture data preprocessing or using matrix operations to calculate distance metrics between faces.

OpenCV:

(Open Source Computer Vision Library) is an open-source computer vision and machine learning software library that can be used to develop applications that involve image and video processing. . In addition to other things, it is frequently used for tracking, object detection, and facial recognition.

IoT (Internet of Things) devices and machine learning algorithms are used to recognise faces using smart face recognition technology. This can be done by employing IoT devices like cameras to take pictures or videos of people, and then processing those pictures or videos using machine learning algorithms to identify the people in the pictures or videos. IoT devices can be utilized to implement various facial recognition techniques and algorithms, which is one way to approach writing a research paper on this subject. This may involve well-known computer vision methods like Eigenfaces, Local Binary Patterns (LBP), and Histogram of Oriented Gradients (HOG), among others Convolutional Neural Networks (CNNs) and Siamese networks are two examples of more recent deep learning-based Methodologies.

Implementation

Here, We completed the project implementation by gathering all necessary hardware and software tools. Combining them all, we incorporate Esp-32 and FTDI to do facial recognition, and Arduino and the OpenCV library are used to create a surveillance system.

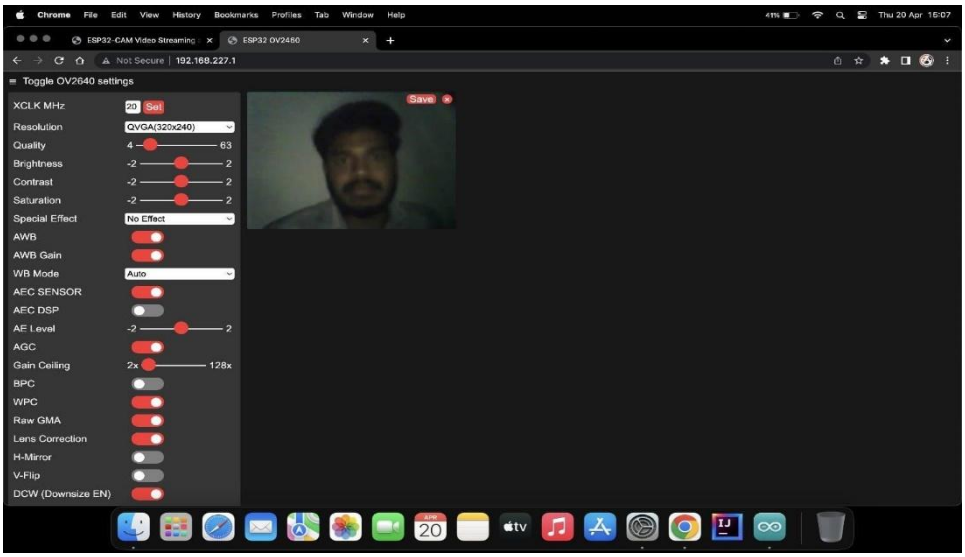

Fig.-5.1.1

Programming using Arduino IDE for Face Recognition Using the Arduino IDE for Face Recognition

- The Arduino IDE must first be configured properly in order for our automobile to run.
- To upload a json file in accordance with your board, go to board management.
- Visit Sketch to obtain the Arduino Uno library.
- Then select the board Arduino Uno under Tools.
- Select the frequency and baud rate.
- Copy the code, paste it in the Arduino IDE, and then compile it.

Working Principle:

The facial recognition method upon which our proposed system is based consists of the following steps.

- We'll start by turning on and configuring the Arduino IDE with ESP-32 and FTDI hardware.

- It will then decide if it is connected to a power source after that. If it is connected, it will start gathering data by using face recognition technology to take pictures of the immediate area, extract the image, and compare the extracted image to the database that is already present in the system.
- Using.csv sheets, it will quickly send an alert to the appropriate authorities, including the name and time, if a match is found with a related database.
- Therefore, the same process will continue until a face and database match is made.Results

We put the suggested strategy into practice and got the intended outcome. As you can see, face recognition has been carried out, and the results will be displayed in a.csv file sheet using the Haar cascade, which recognises objects in images.

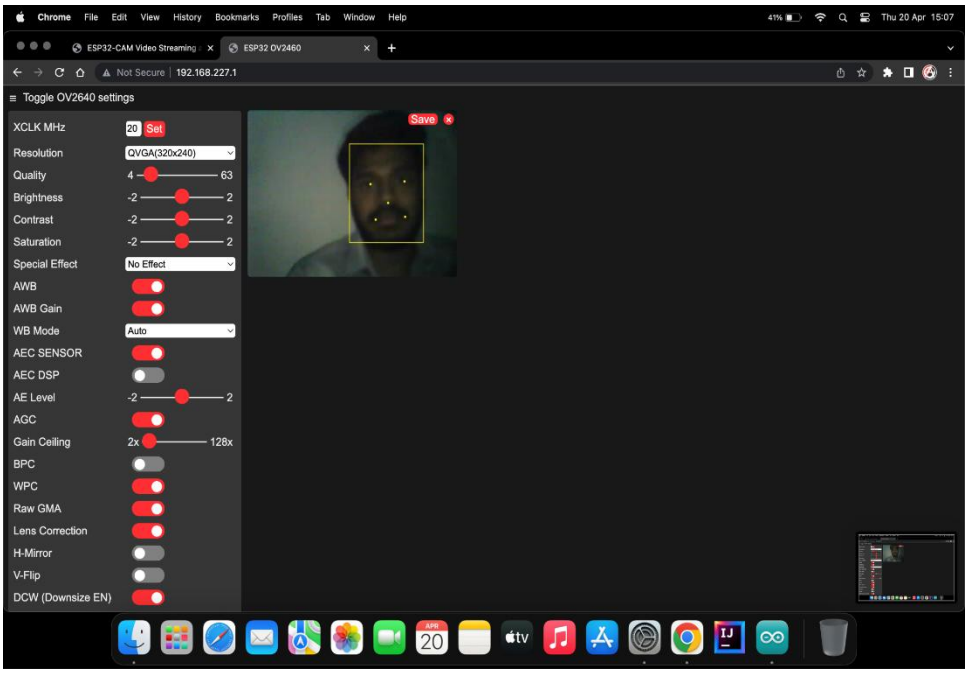

Fig.-6.1.1

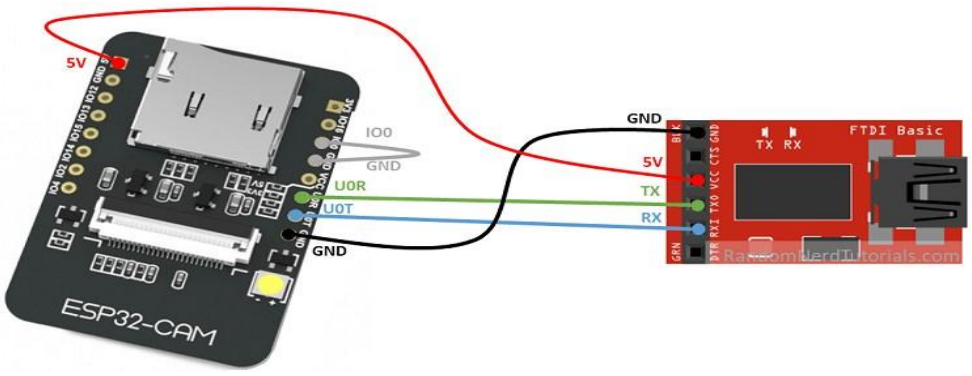

Fig.-6.1.2

6.2 Conclusion:

Face recognition-based human or criminal detection systems can thus be shown to be reliable and safe. face detection and identification using a certain approach and configuration using a variety of hardware and software, such as OpenCV. Similar to this, the Arduino ide, ESP-32 and FTDI is successfully used to extract the image from its surroundings and to match the image with the following database that has been stored. It offers a greater recognition rate with a reduced false rate.The system can be used as a security surveillance system, and by adding a Raspberry Pi infrared camera module, its identification rate can be raised.

So, it can be demonstrated that face recognition-based human or criminal detection systems are safe and effective. Utilizing a specific methodology and configuration with a range of hardware and software, such as OpenCV, for face detection and identification. In a similar manner, the Raspberry Pi 3 board is effectively employed to extract the image from its surroundings and to match the image from the next stored database. With a lower false rate, it provides a higher recognition rate. The system can be utilized as a security surveillance system and its recognition rate can be increased by using Raspberry Pi Infra-Red camera module further.

6.3 Future Scope:

If the demand for face recognition technology rises, it will be useful to everyone in the near future. The proposed technology will then be useful for more than only criminal detection. It will function effectively with projects like tracking attendance, home security, business, retail, and parking, among others. Additional features, such as a location with Ip address tracking system to know the precise location where a criminal was discovered or even the ability to send an alert to the closest crime investigation department looking for criminals, can be added if it is approved and proves to be effective.

REFERENCES

1. [TP91] *M. Turk and A. Pentland*. **Eigenfaces for recognition**. Journal of Cognitive Neuroscience, 3(1):71-86, 1991.
2. [ZKC+98] *W. Zhao, A. Krishnaswamy, R. Chellappa, D. Swets and J. Weng*.**Discriminant analysis of principal components for face recognition**, pages 73-85. Springer Verlag Berlin, 1998.
3. [GHW12] *M. Günther, D. Haufe and R.P. Würtz*. **Face recognition with disparity corrected Gabor phase differences**. In Artificial neural networks and machine learning, volume 7552 of Lecture Notes in Computer Science, pages 411-418. 9/2012.
4. Khan, M. H. Javed, E. Ahmed, S. A. A. Shah and S. U. Ali, "Facial Recognition usingConvolutional Neural Networks and Implementation on Smart Glasses," 2019 International Conference on Information Science and Communication Technology (ICISCT), 2019, pp. 1-6, doi: 10.1109/CISCT.2019.8777442.
5. Mehedi Masud, Ghulam Muhammad, Hesham Alhumyani, Sultan S Alshamrani, Omar Cheikhrouhou, Saleh Ibrahim, M. Shamim Hossain,
6. Deep learning-based intelligent face recognition in IoT-cloud environment,
7. Computer Communications, Volume 152, 2020, Pages 215-222, ISSN 0140- 3664.
8. Bhatti, K., Mughal, L., Khuhawar, F., &Memon, S. (2018). Smart attendance management system using face recognition. EAI Endorsed Transactions on Creative Technologies, 5(17).
9. Kumar, P. M., Gandhi, U., Varatharajan, R., Manogaran, G., & Vadivel, T. (2019). Intelligent face recognition and navigation system using neural learning for smart security in the Internet of Things. Cluster Computing, 22(4), 7733-7744.
10. Agarwal, L., Mukim, M., Sharma, H., Bhandari, A., & Mishra, A. (2021, March). Face recognition based smart and robust attendance monitoring using deep CNN. In 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 699-704). IEEE.
11. Kumar, T. A., Rajmohan, R., Pavithra, M., Ajagbe, S. A., Hodhod, R., & Gaber, T. (2022). Automatic face mask detection system in public transportation in smart cities using IoT and deep learning. Electronics, 11(6), 904.
12. Atik, M. E., & Duran, Z. (2020, October). Deep learning-based 3d face recognition using derived features from point cloud. In The Proceedings of the Third International Conference on Smart City Applications (pp. 797-808). Springer, Cham.