AUTOMATIC ATTENDANCE MANAGEMENT SYSTEM

Abstract**- Facial recognition technology plays a crucial role in various applications, from enhancing security at banks and organizations to streamlining attendance tracking in public gatherings and educational institutions. Traditional methods of attendance marking, such as signatures, names, and biometrics, can be time-consuming and error-prone. To address these challenges, a smart attendance system is proposed, leveraging Deep Learning, Convolutional Neural Networks (CNN), and the OpenCV library in Python for efficient face detection and recognition [1].**

The system utilizes advanced algorithms, including Eigen faces and fisher faces [4], to recognize faces accurately. While deep learning models excel with large datasets, they may not perform optimally with few samples. By comparing input faces with images in the dataset, the system automatically updates recognized names and timestamps into a CSV file, which is then sent to the respective organization's head. Additionally, the system allows users to upload a single photo or a group photo, and it returns matched photos as output using a CNN. This feature enhances the system's flexibility and usability, providing users with a convenient way to identify and track individuals in various scenarios. [2].

Keywords- CNN algorithm, OpenCV, Attendance Tracking, Deep Learning, Dataset Preparation

I. INTRODUCTION

Facial recognition technology has revolutionized security and attendance tracking systems, offering a more efficient and accurate alternative to traditional methods such as signatures and biometrics [2]. This technology finds extensive application in banks, organizations, public gatherings, and educational institutions, where maintaining security and tracking attendance are paramount.

To enhance the capabilities of existing systems, a smart attendance system is proposed, leveraging Deep Learning, Convolutional Neural Networks (CNN), and the OpenCV [3] library in Python. This system utilizes advanced algorithms like Eigen faces and fisher faces for precise face detection and recognition, ensuring accurate attendance marking.

While deep learning models excel with large datasets, they often struggle with few samples [4]. To overcome this limitation, the system compares input faces with images in the dataset, automatically updating recognized names and timestamps into a CSV file. Additionally, the system allows users to upload a single photo or a group photo, returning matched photos as output using a CNN [5]. This feature enhances the system's flexibility [6] and usability, providing users with a convenient and reliable method for identifying and tracking individuals.

II. LITERATURE SURVEY

P.N. Belhumeur, J.P. Hespanha, & D.J. Kriegman - Known for their work on "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," which discusses the application of PCA and LDA in facial recognition technology to enhance identification accuracy.

Joy Buolamwini & Timnit Gebru - Authored "Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification," highlighting biases in commercial facial recognition technologies and advocating for more inclusive datasets.

Clare Garvie, Alvaro Bedoya, & Jonathan Frankle - Co-authored "The Perpetual Line-Up: Unregulated Police Face Recognition in America," a comprehensive report on the use of facial recognition by law enforcement in the United States, raising concerns about privacy and civil liberties.

Alex Krizhevsky, Ilya Sutskever, & Geoffrey E. Hinton - Their groundbreaking paper "ImageNet Classification with Deep Convolutional Neural Networks" significantly advanced the field of deep learning, particularly in image recognition tasks.

Over the past few years, modern researchers have been developing solutions for various problems involving face detection and recognition. According to various surveys, a few models have been proposed. Aiman proposed a model for face detection using a deep learning neural network with a small dataset of images. The images in the dataset are applied with Gaussian and Poisson noise, which results in doubling the dataset. CNN contains the RELU layer, which converts the images into a better form, achieving a good success rate.

My proposal aligns with the trend of using facial recognition for attendance tracking, which has been widely studied and implemented in recent years. The use of deep learning and CNNs for face detection and recognition is well-established in the literature due to their ability to learn complex patterns in images. The automatic update of recognized names and timestamps into a CSV file aligns with the goal of streamlining attendance tracking processes. This feature has been implemented in various attendance management systems using facial recognition technology. The system is implemented using the OpenCV library in Python for face detection and recognition. OpenCV provides a robust and efficient framework for image processing tasks, making it ideal for real-time applications such as attendance tracking. The system allows users to upload single or group photos, which are then matched against images in the dataset using the trained CNN model. Allowing users to upload a single photo or a group photo for matching faces is a useful feature that enhances the system's usability. This feature is often implemented using CNNs for face detection and recognition in real-world applications. My proposed system focuses on using a CNN to extract features from input images (either single or group photos) and then matching these features with images in the dataset to find the best matches. This approach is commonly used in facial recognition systems to compare and identify individuals across different images.

COMPUTER VISION:

Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps [4].

OpenCV: is the huge open-source library for the computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even handwriting of a human. When it integrated with various libraries, such as Numpy, python is capable of processing the OpenCV array structure for analysis. To identify image pattern and its various features we use vector space and perform mathematical operations on these features. [6,7]

CNN ARCHITECTURE:

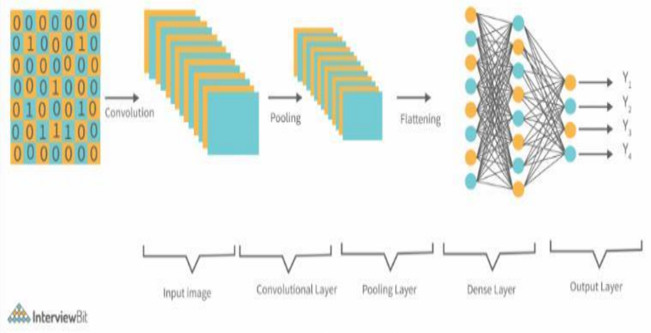


Fig 2: CNN Architecture

Fig 2 shows the overview of each layer in a typical Convolutional Neural Network (CNN) architecture:

Input 2Layer: This layer represents the input image or a batch of images. Each image is typically represented as a 3D array (Height x width x channels), where the channels can be RGB (3 channels) or grayscale (1 channel).

Convolutional Layer: The convolutional layer applies a set Of filters to the input image(s). Each filter is a small matrix that Slides over the input image(s) to produce a feature map. These filters help detect various features in the images, such as edges, textures, or patterns.

Pooling Layer: The pooling layer reduces the spatial dimensions of the feature maps generated by the convolutional layer. It helps in reducing the computational complexity of the network and makes the model more robust to variations in the input images. Common pooling operations include max pooling and average pooling.

Dense (Fully Connected) Layer:The dense layer is a traditional neural network layer where each neuron is connected to every neuron in the previous layer. This layer is typically used at the end of the CNN architecture to classify the features extracted from the previous layers into different classes

Output Layer: The output layer produces the final output of the CNN, which could be class probabilities (in the case of Classification tasks) or continuous values (in the case of regression tasks). The number of neurons in the output layer depends on the number of classes in the classification task or the number of output values in the regression task.

CNN architecture consists of layers designed to extract features from input images and classify them into different categories. The convolutional and pooling layers extract and reduce features, while the dense layer performs the classification based on these features, with the output layer providing the final prediction. [12] .

WORK FLOW:

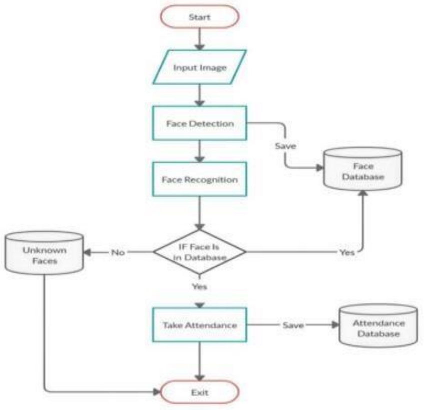


Fig 3: Work of Face Recognition

The fig 3 shows the face recognition it help to guide the implementation process in step by step.

IMPLEMENTATION:

OpenCV is a powerful tool for image processing and computer vision tasks, making it well-suited for developing an attendance management system based on facial recognition.

Using Flask for my interface is a good as it will allow you to create a web-based interface for your attendance management system. Flask is a lightweight and flexible framework, making it easy to integrate with my OpenCV-based backend.

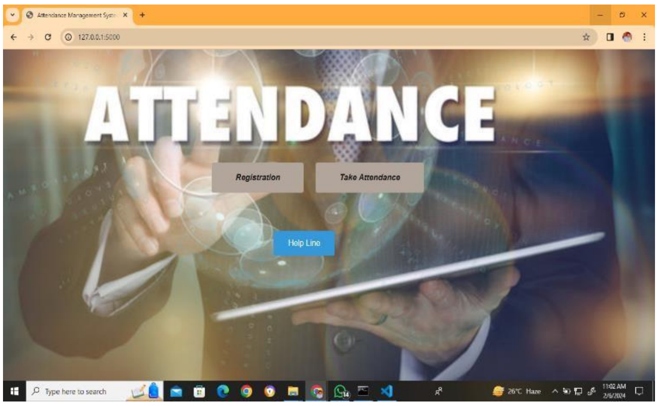


Fig 3: web interface

The fig 3 shows web interface contains three features: Registration, Attendance Taking, and Help.

The when the user clicks on the "Registration" button, a popup message appears prompting them to enter a password.

This feature ensures that only authorized users, such as the admin or main person, can register new users. If the admin enters the correct password, the system navigates to the registration page. Otherwise, it displays a message indicating that the password is incorrect and prompts the admin to enter the correct password [11].

The registration process, users provide their names and IDs, and upon clicking the "Save Profile" button, the system opens the camera to capture several face photos in different poses and expressions. These photos are then saved in a designated folder. Finally, a message is displayed confirming that the profile has been successfully saved.

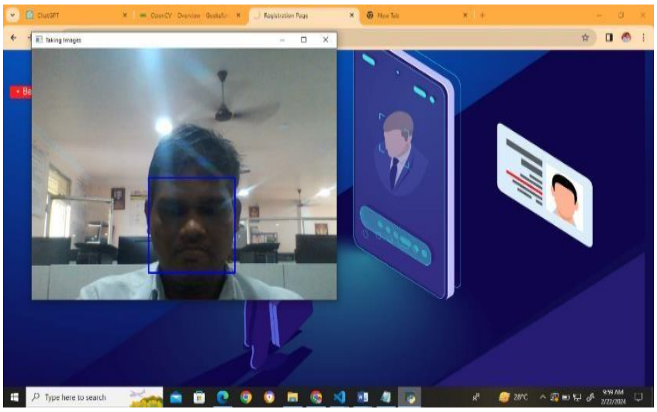


Fig 4: Image’s capturing

Fig 4 shows after saving the image, the system will perform face recognition to take attendance. It will open and identify the faces, displaying the person based on our data otherwise, it will show that the face was not recognized. After recognizing faces, the camera will turn off automatically, and the recognized faces' names and IDs will be stored in one Excel sheet, including the time and date. When we click the download button, the Excel sheet will be downloaded to PC.

The capturing the attendance and save the attendance it saved in excel sheet.

WORK FLOW OF IMAGE MATCHING:

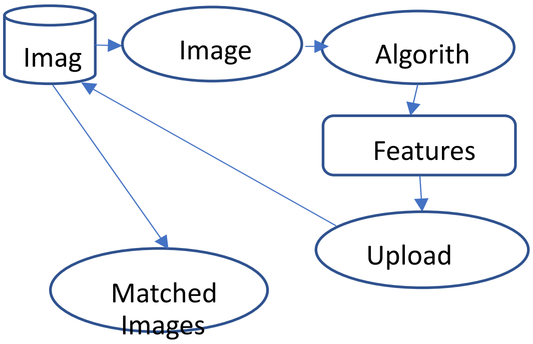


Fig 5: Matching image work flow

IMAGE PREPROCESSING:

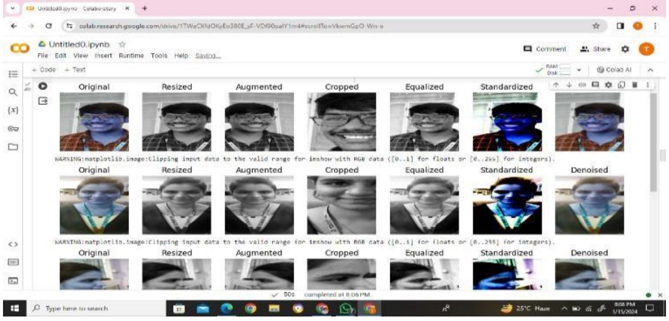
Fig 8 shows the using deep learning pre-processing methods .we perform the image pre-processing on image dataset. Like

A. **Resizing:** This refers to changing the dimensions of an image. It can be done to fit a certain size requirement or to prepare images for processing in a machine learning model, which often requires images to be of a consistent size.

B. **Normalization:** Normalization is the process of adjusting the values of an image to a standard range. In the case of images, this usually means scaling the pixel values to be between 0 and 1, or -1 and 1. Normalization helps in reducing the variation in pixel values and can improve the performance of machine learning models [12].

C. **Rescale:** Rescaling is similar to normalization but involves scaling the pixel values by a certain factor. For example, rescaling by 1/255 would convert pixel values from the range 0-255 to the range 0-1.

D. **Augmentation:** Data augmentation is a technique used to artificially expand the size of a dataset by creating modified versions of images in the dataset. This can include flipping, rotating, cropping, or changing the brightness of images. Data augmentation is commonly used in deep learning to improve the robustness of models and prevent overfitting.



```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
```

III. TRAIN CNN MODEL

During the training of a Convolutional Neural Network (CNN) algorithm, the model's accuracy typically improves over time as it learns from the dataset. This improvement is often visualized in a figure where the x-axis represents the number of epochs or iterations over the training dataset, and the y-axis represents the model's accuracy.

In your case, the figure below shows the accuracy trend of the CNN algorithm during training, with the x-axis indicating the number of epochs and the y-axis indicating the accuracy percentage. The figure illustrates how the model's accuracy gradually increases with each epoch, indicating that the model is learning and improving its performance over time. The final accuracy achieved, such as 71.34%, reflects the model's ability to correctly classify images in the test set after training.

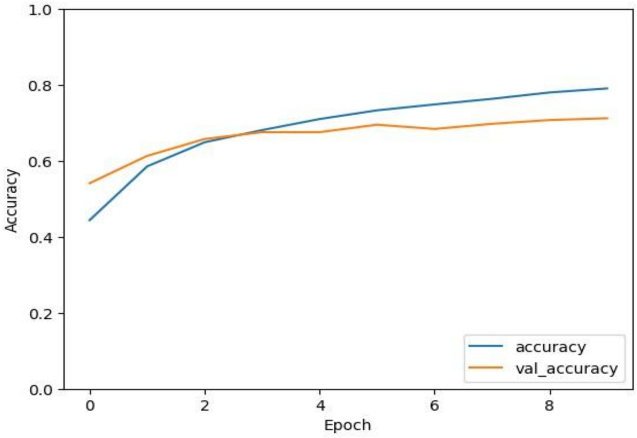
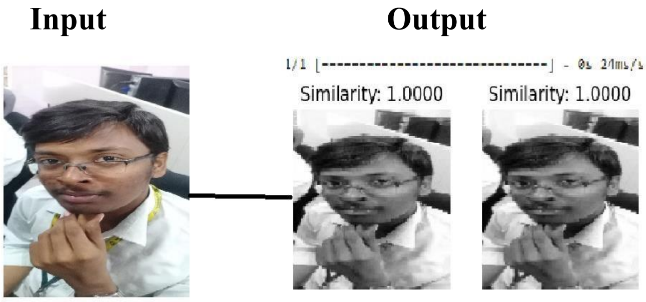


Fig 6: CNN training Accuracy graph

Understanding the accuracy trend is crucial for evaluating the performance of the CNN algorithm[12] and identifying potential issues such as overfitting or under fitting. Fig 8 shows analyzing the accuracy trend, you can make informed decisions about the model architecture, hyper parameters, and training process to optimize the model's performance.

To save the trained model with the name "best_match.h5," you can use the model's serialization functionality. This allows you to save the model's architecture, weights, and training configuration to a file, which can then be loaded later for inference or further training. It's important to use separate training and test datasets to evaluate the model's performance accurately. The training dataset is used to train the model, while the test dataset is used to evaluate its performance on unseen data.

IV. RESULT



```
1/1 [==============================] - 0s 24ms/s
```

Fig 7: Single photo matching

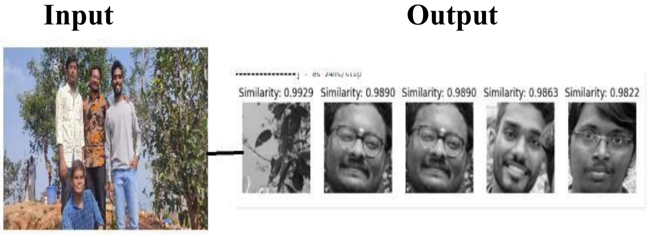


Fig 8: Group photo Matching

Fig 7 & fig 8 shows the single and group photo matching using CNN algorithm.

V. CONCLUSION AND FUTURE SCOPE

The proposed model has the capability of detecting and recognizing different faces and images from the camera. The face recognized matching purpose we using another module OpenCV. It's a powerful library for computer vision tasks, and it should work well for your facial recognition system. The data set which contains the images are pre-trained and tested using deep learning so that the input images would be well detected. This method is secure enough, reliable and available for use. Further CNN adds robustness to the model and using this approach of training data, 71.34% recognition rate has been achieved. Deep learning has advantage over machine learning for other face recognition techniques. The resultant of this entire process is nothing but creating an attendance marking system in which the unique id, name and some more details of the recognized faces could be entered automatically into a CSV file. In further updating this attendance system can be taken towards web development by creating a website and marking the attendance of the people automatically into the website of the organization so that there would be no need to update or mark attendance manually.

Additionally, the model's ability to find the best match photos from the dataset when uploading single or group photos enhances its utility and effectiveness.

Presently this project is developed using Flask python and is successfully running on web. In the future upation, a mobile application will be developed in which each and every student are given access with unique login details so that they can track their status of attendance from anywhere round the globe.

REFERANCES:

[1] H. Yang and X. Han, "Face Recognition Attendance System Based on Real-Time Video Processing", *IEEE Access*, vol. 8, pp. 159143-159150, 2020.

[2] D. M. Prasanna and C. G. Reddy, "Development of Real Time Face Recognition System Using OpenCV", *International Research Journal of Engineering and Technology*.

[3] Siti Ummi Masruroh et al., "NFC Based Mobile Attendence System with Facial Authorization on Raspberry Pi and Cloud Server", *2018 6th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1-6, 2018.

[4] J. Alghamdi, R. Alharthi, R. Alghamdi, W. Alsubaie, R. Alsubaie, D. Alqahtani, et al., "A survey on face recognition algorithms", *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*, pp. 1-5, 2020.

[5] S. M. Anzar, N. P. Subheesh, A. Panthakkan, S. Malayil and H. A. Ahmad, "Random interval attendance management system (RIAMS): A novel multimodal approach for post-COVID virtual learning", *IEEE Access*, vol. 9, pp. 91001-91016, 2021.

[6] J. W. D'Souza, S. Jothi and A. Chandrasekar, "Automated attendance marking and management system by facial recognition using histogram", *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*, pp. 66-69, 2019.

[7] T. Sharanya, U. Kasturi, P. Sucharith, T. Mahesh and V. Dhivya, "Online attendance using facial recognition", *Int. J. Eng. Res*, vol. 9, no. 06, pp. 202-207, 2020.

[8] N. Yasmeen, K. K. R. C and S. Doss, "Intelligent systems powered hourly attendance capturing system", *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1536-1541, 2023.

[9] R. S. K. A. A. R and C. K. Kannan, "Automatic attendance monitoring system using LBPH and HAAR algorithm", *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, vol. 1, pp. 1604-1608, 2023.

[10] K. Vignesh, A. Abirami, P. Manideep, K. Vasu, K. Rakesh and S. V. Vardhan, "Smart attendance system using deep learning", *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 1081-1087, 2023;

[11] NING ZHU, ZEKUAN YU and CAIXIA KOU, "A New Deep Neural Architecture Search Pipeline for Face Recognition", *IEEE Access*, 2020.

[12] Syam Kakarla, Priyaranjan Gangula, M. Sai Rahul, C. Sai Charan Singh and T. Hitendra Sarma, "IEEE-HYDCON Smart Attendance Management System Based on Face Recognition Using CNN", *2020 IEEE-HYDCON*, 2020.