JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AUTOMATED FACE RECOGNITION AND ATTENDANCE MANAGEMENT SYSTEM USING PYTHON OPENCV AND CONVOLUTIONAL NEURAL NETWORK(CNN)

Mrs.M.Sasipriya M.E[1], R.Gokilan[2], K.Nithish[3], S.Vasanth Kumar[4],
[1] Assistant Professor, [2][3][4] Students , Department Of Information Technology ,
K.S.R. College Of Engineering,Tiruchengode,Tamilnadu, India.

Abstract:

In today's technologically advanced era, the demand for efficient and automated attendance management systems is paramount. The system can scan faces in real time using a webcam. It employs the face recognition library to compare face encodings and identify individuals from a preloaded set of images. The system keeps track of attendance by marking the entry and exit times of recognized individuals. It maintains an attendance CSV file with details such as name, reg number, status (present or absent), entry time, exit time, late entry, time spent, and attendance percentage. The system can send an email with an attached attendance report every 50 minutes with day attendance, hour attendance, and the complete attendance percentage to date. This feature enhances communication and allows for easy sharing of attendance records with management. The GUI provides attractive buttons for functionalities like face scanning, adding new faces, deleting faces, and displaying scanned images.

Keywords: computer vision; face recognition; image processing; attendance management system.

I. INTRODUCTION

A.Face Recognition

Facial recognition technology has gained widespread attention in recent years due to its applications in security, surveillance, and personal identification systems. This project aims to develop a Facial Recognition System using Python, leveraging the power of computer vision libraries.

The system utilizes the face_ recognition library for face detection and recognition tasks, OpenCV for image processing, and TensorFlow for deep learning-based facial recognition model training and inference. Additionally, tkinter is employed for building a user-friendly graphical interface to interact with the system.

This facial Recognition System offers a robust and versatile solution for various industries seeking to leverage facial recognition technology for enhanced security, efficiency, and convenience.

B.Computer Vision

The goal of the artificial intelligence and computer science fields of computer vision is to enable computers to interpret and comprehend visual data from the outside environment. It entails the creation of methods and algorithms that enable computers to interpret, evaluate, and derive meaning from visual data, including photos and movies.

By enabling computers to detect and understand visual input like humans, computer vision aims to both mimic and improve human vision skills. This covers tasks including tracking, segmentation, picture classification, object identification, recognition, scene comprehension, and 3D reconstruction.

Classifying photos into preset classes or categories is the process of image classification. Convolutional neural networks, or CNNs, have become an extremely effective tool for image classification applications, producing cutting-edge outcomes across a range of applications. Object tracking is the process of tracking an object's movement throughout a video sequence and over time.

C.Time Awareness

Time awareness in the facial recognition project encompasses various aspects that are difficult to the system's functionality and effectiveness. Firstly, the system needs to accurately timestamp attendance records, capturing the entry and exit times of individuals recognized by the facial recognition system. This ensures that attendance tracking is not only precise but also provides insights into the duration of individuals' presence, aiding in monitoring and analysis.

Efficient algorithms and techniques are required to minimize processing time, enabling swift responses to user actions and ensuring smooth operation of the system. Moreover, training the facial recognition model involves time considerations, as the process can be time-consuming, particularly for large datasets and complex model architectures. Utilizing efficient training strategies and hardware acceleration can help mitigate training time while maintaining model accuracy.

II. Literature Review

[1] The efficacy of conventional face recognition techniques, such as Eigenfaces, Fisherfaces, and Local Binary Patterns (LBP), in recognizing people based only on their facial traits has been thoroughly investigated. With the advent of deep learning and Convolutional Neural Networks (CNNs), face recognition has undergone a radical change. These CNNs do not require human feature extraction since they can learn intricate hierarchical representations straight from raw pixel input. VGGNet, ResNet, and Siamese network are a few well-known CNN designs for facial recognition.

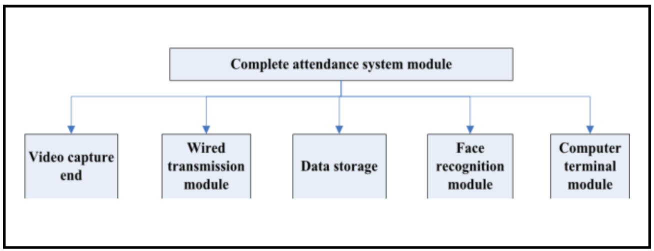
[2] Automated attendance management systems leverage various technologies, including RFID, biometrics, and computer vision, to automate the process of attendance tracking. Biometric-based systems, such as fingerprint recognition and iris scanning, offer high accuracy but may pose privacy concerns and require physical contact. Computer vision-based systems, including face recognition, provide a non-invasive and contactless alternative for attendance tracking, making them suitable for diverse environments and applications.

[3] Python's ease of use, adaptability, and large library have made it a popular choice for computer vision applications. A popular open-source toolkit for computer vision and image processing applications, OpenCV (Open Source Computer Vision toolkit) offers a wide range of features for tasks like object identification, feature detection, and picture editing.

[4] In a variety of computer vision applications, such as object detection, picture categorization, and face recognition, CNNs have shown impressive performance. CNN-based face recognition models, which are trained on extensive datasets like LFW (Labeled Faces in the Wild) and CelebA, often comprise many convolutional layers, succeeded by fully connected layers. Deep learning models with minimal training data have been adapted to specific face recognition tasks through the use of transfer learning approaches, such as fine-tuning pre-trained CNN models.

[5] Combining CNN with OpenCV enables real-time face recognition and tracking in video streams, facilitating applications such as surveillance, access control, and attendance management. OpenCV provides functionalities for face detection, alignment, and feature extraction, which complement the deep learning capabilities of CNN models.

[6] CSV files are like lists of items separated by commas. Each line represents a row of data, and the commas split up the different pieces of information. They're simple and work with many programs. Excel files, on the other hand, are like digital notebooks with different pages called sheets. Each sheet is like a grid where you can organize your data into rows and columns. Excel lets you do more things with your data, like formatting it, doing calculations, and making charts. While Excel files are mainly used with Microsoft Excel, other programs can also work with them. So, CSV is basic and straightforward, while Excel gives you more options for working with your data.



[7] The graphical user interface (GUI) of the facial recognition system project is designed to provide an intuitive platform for users to interact with the system's functionalities. The main window is titled "Facial Recognition System" and features a layout created using the custom winter library. Within this layout, several buttons are strategically positioned to offer distinct functionalities. These include options such as initiating face scanning through the webcam, displaying known images for facial recognition, capturing new images to add to the system, and deleting existing images from the database. Additionally, there's a button to open an informational file or image about the project

[8] the system collects a bunch of face images from a specific folder. These images are used as a reference for identification. Then, using a special neural network called a Convolutional Neural Network (CNN), the system extracts unique features from each face image, sort of like creating a special fingerprint for each face. When you show a new image to the system, it also extracts features from it and compares them to the features of the faces it knows. If the new image's features are similar to any of the known faces, the system says it's found a match. If not, it says it doesn't recognize the face. After recognizing faces, the system updates attendance records and creates reports.

III. Existing System

The existing assistant system is designed with a primary focus on real-time video face recognition for attendance systems, including face detection, feature extraction, recognition methods, and system architecture.

Capturing and Preprocessing the Face Image: The first step is to capture a new image of the person's face using a webcam. This image is then preprocessed to ensure it's suitable for face recognition. Preprocessing may involve converting the image to grayscale, resizing it, and performing any necessary normalization or enhancement to improve the quality of the image.

Encoding the Face Image: Once the face image is captured and preprocessed, it needs to be encoded to generate a unique representation of the person's face. Face encoding techniques typically involve extracting features from the face image and transforming them into a numerical representation, such as a vector. This encoding is what the face recognition model uses to identify individuals.

Updating the Dataset: The next step is to update the existing dataset with the newly encoded face image. The dataset usually consists of a collection of face encodings along with their corresponding IDs or labels. In this step, the new encoding is added to the dataset, associating it with the ID of the person whose face was captured.

Retraining the Face Recognition Model: Once the dataset is updated with the new face encoding, the face recognition model needs to be retrained using the updated dataset. This involves feeding the model with the training data (face encodings and corresponding IDs) and adjusting its parameters to improve its ability to recognize faces accurately. Training typically involves using machine learning algorithms such as Support Vector Machines (SVM) or k-nearest Neighbors (k-NN) to learn patterns from the data and make predictions.

Validation and Testing: After retraining the model, it's essential to validate its performance to ensure that it can accurately recognize faces, including the newly added ones. This may involve testing the model with a separate set of test data or evaluating its performance on real-world face recognition tasks.

IV. PROPOSED SYSTEM

A. Managing Large Data

One effective step to optimize the code for managing large data in the Facial Recognition System is to implement lazy loading or streaming techniques for processing images in batches. Lazy loading involves loading data only when it's needed, rather than loading everything into memory upfront. In the context of the Facial Recognition System, this means loading images from disk only when they are about to be processed, rather than loading the entire dataset into memory.

Streaming techniques involve processing data in sequential chunks or batches, rather than processing the entire dataset in one go. This allows the system to handle large volumes of data more efficiently by reducing memory overhead and enabling continuous processing without exhausting system resources. By implementing lazy loading and streaming techniques, the Facial Recognition System can efficiently process large datasets without encountering memory limitations or performance issues.

B. Email Notifications

After each class, an email is automatically sent to the teacher and head of department, detailing who attended and who didn't. The email includes class specifics like the time and instructor, along with individual student attendance records showing when they arrived, left, and how long they stayed. It also calculates the percentage of students present. This system works seamlessly, checking attendance records, composing the email content, and sending it out effortlessly. We ensure the emails are clear and easy to understand, allowing teachers and heads of department to quickly grasp attendance trends. Before implementing this system, thorough testing is conducted to iron out any issues, ensuring smooth operation.

C. Implementing CNN Algorithm

Convolutional, pooling, and fully connected layers comprise the architecture of a Convolutional Neural Network (CNN), which is constructed through definition. Fully connected layers process flattened output for predictions, pooling layers decrease feature maps while maintaining crucial information, and convolutional layers extract features from input data. An optimizer, a loss function, and assessment metrics like Adam or stochastic gradient descent are included in the compilation of the model. The model is trained using a labeled dataset, and gradient descent and backpropagation are used to modify parameters to minimize the loss function. Lastly, a different test dataset is used to evaluate the model's performance and capacity for generalization. Creating an efficient CNN model for a given job requires experimenting with various architectures and fine-tuning hyperparameters.

V. SYSTEM DESIGN SYSTEM

A.User Interface

Examining the user interface as a whole and how it varies from the more conventional graphical user interface that most recent apps employ is crucial.

A.1.Graphical User Interface

The most often utilized type of interface nowadays is a graphical one. The user may interact with machines more quickly and easily than in the past by using graphical symbols and visual cues.

A.2.User Interface in Face recognizer

The graphical user interface (GUI) of your facial recognition system is like the control panel of the application, where you can see and interact with different features. When you open the application, it loads images of known faces from a folder and prepares them for recognition. If everything is set up correctly, it loads a special "brain" called a model, which helps in recognizing faces. The GUI has buttons for different tasks like scanning faces using a webcam, adding new faces, deleting faces, and viewing known faces. For example, when you click the "Add a new face" button, it takes a picture using your webcam and saves it in the system. You can also view all the known faces and delete them if needed. The GUI also helps you manage attendance by marking who attended based on recognized faces. Additionally, there are buttons for tasks like viewing attendance reports and learning more about the application. Overall, the GUI makes it easy for you to use the facial recognition system without needing to write any code yourself.

V. IMPLEMENTATION

A.Modules Description

A.1.Face_recognition:

This module is vital for face detection and recognition tasks. It offers functions for encoding faces, detecting facial features, and comparing faces against a database. This module is fundamental for the core functionality of your facial recognition system, as it provides the necessary algorithms and tools to identify individuals from images or video streams.

A.2.numpy(as np):

NumPy is indispensable for scientific computing and array manipulation in Python. In your code, it likely plays a critical role in handling image data and performing mathematical operations on arrays. NumPy's array operations are efficient and essential for processing large datasets, making it a fundamental component for tasks like image preprocessing and feature extraction.

A.3.cv2:

OpenCV (Open Source Computer Vision Library) is essential for image processing and computer vision tasks. In your code, it facilitates various operations such as reading, displaying, and manipulating images. Additionally, OpenCV provides functionalities for webcam access, which is crucial for real-time face detection and recognition applications

A.4.CNN Model Development:

The Convolutional Neural Network (CNN) model is developed to perform face recognition tasks using the collected and preprocessed data. This involves designing the architecture of the CNN model, including the number of layers, types of layers (convolutional, pooling, fully connected), and activation functions. The model is trained on the preprocessed dataset using appropriate optimization algorithms and evaluation metrics to achieve high accuracy in face recognition.

B. Implementation Details:

B.1.Data Preprocessing:

The facial images collected for training undergo preprocessing to ensure uniformity and quality. Common preprocessing steps include:

- Resizing: Resize all images to a standard size (e.g., 128x128 pixels) to maintain consistency.
- Normalization: Normalize pixel values to a common scale (e.g., [0, 1]) to enhance convergence during training.
- Augmentation: To improve variety and resilience, apply changes to the dataset, such as rotation, scaling, and flipping.

B.2.Training the CNN Model:

- Utilizing optimization methods like stochastic gradient descent (SGD), Adam, or RMSprop, train the CNN model on the preprocessed dataset. While in training, the following factors are taken into account:
- Loss function: To calculate the difference between the predicted and real labels, use an appropriate loss function (categorical cross-entropy).
- Learning rate: Adjust the learning rate dynamically to control the speed of convergence and prevent overshooting.
- Batch size: Determine an appropriate batch size to balance between computational efficiency and convergence speed.
- Number of epochs: Training my model for a sufficient count of epochs to ensure convergence while avoiding overfitting.

B.3.Graphical User Interface:

The GUI is constructed using the **customtkinter** and **tkinter** libraries, providing an intuitive interface for users to interact with the facial recognition system. It includes various buttons for actions such as scanning faces from the webcam, adding or deleting faces, viewing known faces, and managing attendance records. The GUI enhances the user experience and facilitates seamless interaction with the system.

B.4. Face Recognition and Attendance :

When a face is detected and recognized, the system updates the attendance records using the **markAttendance()** function. This function records the entry time of the recognized individual and marks them as present in the attendance record. The attendance data is then saved to a CSV file (**Attendance.csv**) for further analysis and reporting. This feature automates the attendance tracking process, streamlining administrative tasks for users.

C. Technology Used

Artificial Intelligence and facial recognition technologies are utilized to quickly and precisely identify the image that the user is looking for. While making the identification could appear easy. The technologies that are widely utilized include Tensorflow, OpenCV, and Face Recognition Library.

C.1. OpenCV :

A popular open-source computer vision library, OpenCV offers a wide range of tools and capabilities for the process of images and video. In your system, OpenCV is crucial for accessing and manipulating images, performing face detection and recognition, and integrating with webcams for real-time face scanning. Its comprehensive set of features makes it indispensable for various computer vision tasks.

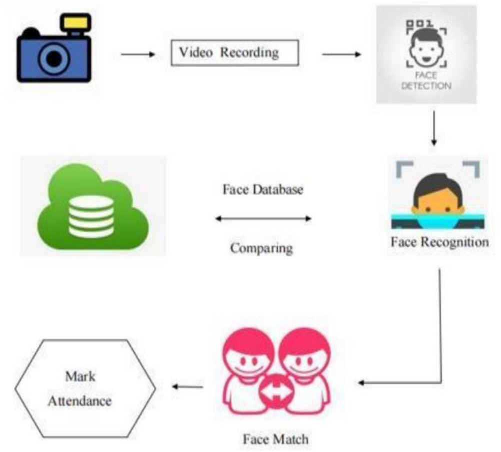
C.2.Artificial Intelligence:

Artificial intelligence (AI) makes computers think and understand on their own, just like humans do. It's used in things like virtual assistants (think Siri or Alexa), self-driving cars, and even medical diagnosis. AI learns from the data it's given, getting better at tasks over time without needing to be explicitly programmed. Deep learning, a type of AI, helps computers recognize patterns and features in things like images or speech. While AI has many exciting applications, it also raises concerns about privacy, fairness, and job displacement. Despite these challenges, AI technology is continuously advancing, offering promising solutions to various problems and shaping the future of many industries.

C.3.TensorFlow:

Google created TensorFlow, an effective open-source deep learning framework that is frequently used for configuring and training deep neural networks. In your system, TensorFlow is employed for loading and potentially training a deep-learning model specifically designed for facial recognition tasks. TensorFlow is a crucial tool for obtaining high accuracy in facial recognition since deep learning models, including convolutional neural networks (CNNs), are excellent at picture identification tasks.

VI. Block Diagram:

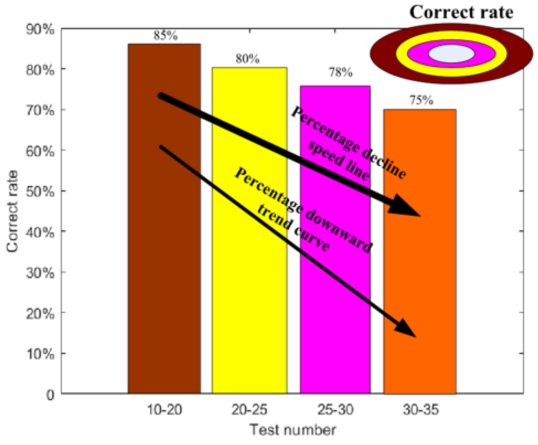


VII. Result Analysis:

Video face rιcognition accuracy	High school A		High school B	
	Test group	Control group	Test group	Control group
Accurate identification (%)	82	94	83	96
Video is blurry (%)	15	0	12	0
Punch failed (%)	3	6	5	4

VIII. Test number and accuracy rate table:

Test number	Correct rate
10～20	85%
20～25	80%
25～30	78%
30～35	75%



XI. *CONCLUSION AND FURTHER SCOPE*

A.1.Conclusion:

- Before this project's development. The old technique of taking attendance had many flaws that generated many problems for most of the institutions.

- As a result, the attendance monitoring system's facial recognition function can guarantee correct attendance taking and fix the shortcomings of the prior setup.

- By handing over all the difficult jobs to the machine, technology used to overcome flaws saves money and minimizes the need for human intervention throughout the process.

- The only expense associated with this approach is having enough room in the database storage to hold all of the faces. Thankfully, micro SD cards are available to make up for the large amount of data.

- The face database in this project has been constructed successfully. Other than that, facial recognition technology is operating efficiently.
- Ultimately, the solution not only fixes issues with the previous model but also makes it easier for users to get the data that was gathered by mailing the attendance sheet to the esteemed professors.

A.2.Further Integration:

Implementing face recognition technology for student attendance management presents several advantages for educational institutions. By deploying robust face recognition algorithms, schools can accurately identify and record students' attendance in real time, eliminating manual processes prone to errors. Integrating this technology into existing attendance management systems streamlines administrative tasks and enhances efficiency. With a secure database storing students' facial templates, the system can quickly verify their presence upon entry, facilitating seamless monitoring of attendance across multiple locations. Additionally, incorporating user-friendly interfaces and feedback mechanisms ensures a positive experience for both students and staff. Privacy concerns can be addressed through strict adherence to regulations and the implementation of encryption techniques to safeguard sensitive facial data. Overall, the integration of face recognition technology revolutionizes student attendance management, promoting automation, accuracy, and efficiency in educational institutions.

X .REFERENCES:

[1] Hao yang1 and Xiaofeng han,``Face Recognition Attendance System Basedon Real-Time Video Processing'' IEEE Access ,vol .8,no .10,pp.1109,July 2020.

[2] C. Ding and D. Tao, ``Trunk-branch ensemble convolutional neural networks for video-based face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 10021014, Apr. 2018.

[3] C. Stoll, R. Palluel-Germain, R. Caldara, J. Lao, M. W. G. Dye, F. Aptel, and O. Pascalis, ``Face recognition is shaped by the use of sign language,'' *J. Deaf Stud. Deaf Educ.*, vol. 23, no. 1, pp. 19, 2018.

[4] K. Taniya, M. Nidhi, and T. Nandini, ``Automated human resource and attendance management system based on real-time face recognition,'' *IJSRSET*, vol. 16, no. 4, pp. 847853, 2016

[5] D. Wu, Y. Tang, G. Lin, and H. Hu, ``Robust face recognition based on significance local directional pattern and deep learning," *J. Optoelectron. Laser*, vol. 27, no. 6, pp. 655661, 2016.

[06] K. Solanki and P. Pittalia, ``Review of face recognition techniques," *Int.J. Comput. Appl.*, vol. 133, no. 12, pp. 20_24, Jan. 2016.

[07] W. Deng, J. Hu, and J. Guo, ``Face recognition via collaborative representation: Its discriminant nature and superposed representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2513_2521, Oct. 2018.

[08] T. Pei, L. Zhang, B.Wang, F. Li, and Z. Zhang, ``Decision pyramid classier for face recognition under complex variations using single sample per person," *Pattern Recognition.*, vol. 64, pp. 305_313, Apr. 2017.

[09] T. Valentine, M. B. Lewis, and P. J. Hills, ``Face-space: A unifying concept in face recognition research," *Quart. J. Experim. Psychol.*, vol. 69, no. 10, pp. 1996_2019, Oct. 2016.

[10] S. P. Mudunuri and S. Biswas, ``Low-resolution face recognition across variations in pose and illumination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 5, pp. 1034_1040, May 2016.

[11] Y. Li, W. Song, and C. Cheng, ``Attendance system of face recognition based on raspberry pi%" *Microcontrollers Embedded Syst. Appl.*, vol. 16, no. 11, pp. 28_30, 34, 2016.

[12] Y. Wu, T. Hassner, K. Kim, G. Medioni, and P. Natarajan, ``Facial land mark detection with tweaked convolutional neural networks," in *Proc.IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp.3067_3074, Dec. 2018.

[13] F. Schroff, D. Kalenichenko, and J. Philbin, ``FaceNet: A uni_ed embedding for face recognition and clustering," in *Proc. IEEE Conf. Computer Vision Pattern Recognit.*, Jun. 2015, pp. 815_823.

[14] C. Kohalli, R. Kulkarni, M. Salimath, M. Hegde, and R. Hongal, ``Smart wireless attendance system," *Int. J. Comput. Sci. Eng.*, vol. 4, no. 10, pp. 131_137, Sep. 2016.

[15] M. M. Islam, M. K. Hasan, M. M. Billah, and M. M. Uddin, ``Development of smartphone-based student attendance system," in *Proc. IEEE Region 10 Humanitarian Technol. Conf. (R10-HTC)*, Dhaka, Bangladesh, Dec. 2017, pp. 230_233.

[16] Z. H. Arif, N. S. Ali, N. A. Zakaria, and M. N. Al-Mhiqani, ``Attendance management system for the educational sector: Critical review," *Int. J. Computer Sci. Mobile Comput.*, vol. 7, no. 8, pp. 60_66, Aug. 2018.