Face Recognition-Based Attendance System Using Python: A Novel Approach to Automated Attendance Management

Shaheer S[1], Adarsh V[2], Afsal Ansari[3], Salman Sunil[4], Nishil A[5]

[1] *Department of Electronics and Communication Engineering, Younus College of Engineering and Technology*
[2] *Department of Computer Science and Engineering, College of Engineering Karunagappally*
[3] *Department of Electronics and Communication Engineering, Younus College of Engineering and Technology*
[4] Department of Electronics and Communication Engineering, Younus College of Engineering and Technology
[5]Department of Electronics and Communication Engineering, Younus College of Engineering and Technology

---***---

Abstract - This study shows a face recognition-based attendance system created with Python. It uses machine learning algorithms to boost accuracy and productivity. Old-school attendance methods often needed to be corrected and took a lot of time. This new system changes that by using real-time facial recognition to do the job. The system relies on a mix of Logistic Regression, Support Vector Machines (SVM), and Random Forest classifiers, with a Voting Classifier to make sure it works as well as possible. The team did a lot of work to clean up facial images and fine-tune settings to make the model more accurate. The results show big improvements in managing attendance, with high accuracy and strong performance in different settings. This system offers a solution that can grow with organizations looking to simplify attendance tracking, reduce human mistakes, and make their operations more effective.

Key Words: Face recognition, attendance system, machine learning, real-time application, Voting Classifier, Python.

1. INTRODUCTION

This study shows a face recognition-based attendance system created with Python. It uses machine-learning algorithms to boost accuracy and productivity. Old-school attendance methods often needed to be corrected and took a lot of time. This new system changes that by using real-time facial recognition to do the job. The system relies on a mix of logistic regression, support vector machines (SVM), and random forest classifiers, with a voting classifier to make sure it works as well as possible. The team did a lot of work to clean up facial images and fine-tune settings to make the model more accurate. The results show big improvements in managing attendance, with high accuracy and strong performance in different settings. This system offers a solution that can grow with organizations looking to simplify attendance tracking, cut down on human mistakes, and make their operations more effective.

In our fast-moving and digital world today, organizations, schools, and many other sectors need strong and dependable systems to manage attendance. The old ways of keeping track of who's present, like calling out names or signing paper lists, take too much time and often lead to mistakes and security issues. These methods can result in wrong records and people can cheat the system, which makes them less useful for ensuring people are held accountable and everything is out in the open.

What's more standard automated ID systems, like fingerprint scanners, ID cards, and passwords, come with their problems. It's hard to get clear fingerprints in some places, people can lose or have their ID cards stolen, and they might forget passwords or have them hacked. These issues show why we need a better, safer, and easier-to-use option.

Face recognition tech has popped up as a promising choice offering a smart and secure way to track attendance. By using unique face features to identify and check people, face recognition systems get rid of many of the problems that come with old-school methods. This tech not only makes attendance records more accurate but also provides a smooth and touch-free experience, which matters when it comes to health and safety worries.

In this study, we show a face recognition attendance system that uses deep learning and convolutional neural networks (CNNs). Our system works in real-time identifying and verifying people as they use it. We've used transfer learning, which lets us adapt pre-trained models for new jobs, to boost how well our CNN models perform. By training our system with a dataset of many types of face images, we've built a strong solution that can be very accurate even when conditions aren't ideal.

This paper will dig into the different parts of our system, including its layout, the machine learning algorithms it uses, and the ways we prepare data and train models. We'll also show what our tests found pointing out how well the system works and how it could be used by many. What we learned from this study proves that face recognition tech is a solid and effective answer to today's attendance tracking needs.

2. LITERATURE SURVEY

Face Recognition-Based Attendance Management System: his paper talks about a system made for schools and companies that uses advanced computer vision to automate attendance. It applies algorithms like Haar Cascade Classifiers and CNNs to detect faces, PCA and deep learning to extract features, and methods such as Eigenfaces to recognize faces. The system can grow, stay secure, and work well with other systems in institutions.

Automated Attendance Management System using Face Recognition: This paper introduces a system that uses face recognition to track attendance fixing problems with old-fashioned methods. The system uses deep learning models to spot and recognize faces in real time giving accurate and quick attendance records while protecting data privacy and allowing the system to grow.

Face Recognition-Based Attendance System Using Raspberry Pi: The system in this paper relies on Raspberry Pi to manage attendance. It's cheap and easy to move around using Haar Cascade Classifiers and Eigenfaces to spot and recognize faces. The system keeps information private and can grow as needed offering schools a way to save money.

Face Recognition-Based Automatic Attendance System using Local Directional Number Pattern: This paper presents a system that applies the Local Directional Number Pattern (LDN) to recognize faces. It performs well under different light conditions and face angles. The system incorporates LDN features to recognize faces, which results in high accuracy and follows privacy rules.

Real-Time Face Recognition-Based Attendance System using Deep Learning: This system uses convolutional neural networks (CNNs) to extract features and classify them, which allows it to recognize faces in real-time. The paper points out that the system can handle tough conditions, which ensures it tracks attendance**.**

FPGA-Based Face Recognition System for Attendance Management: This paper looks into how Field-Programmable Gate Array (FPGA) technology can help with real-time face recognition in attendance tracking. It zeroes in on the way FPGAs can process things side by side, which leads to quick responses and high output. This makes the system a good fit for all sorts of settings.

Smart Attendance System Using Face Recognition:
This paper shows a clever attendance system that uses deep learning methods like CNNs to spot and identify faces. The system aims to boost productivity cut down on office work, and keep people's information safe when tracking attendance. This means it can be useful in many different fields.

3. METHODOLOGY

Our research uses a step-by-step method to create and improve a face recognition model. We break down the process into six main parts:

3.1 Dataset Details

Our study uses a set of face pictures, each linked to a specific identity or attribute tag. We prepare these images beforehand to get face details, like feature vectors, which play a key role in training our model. We split our dataset into two main parts: the independent variables (x), which are the face details we've extracted, and the dependent variables (y), which are the matching tags. Each face description has 128 features showing important aspects of the face structure that we need for accurate recognition.

3.2 Data Loading and Preprocessing

The study kicks off by pulling in the pre-processed dataset from a pickle file. This file comes from an earlier data preprocessing script. The dataset arranged as NumPy arrays or Pandas Data Frames, gets split into independent variables (x) and dependent variables (y). To make sure they work with machine learning models, the face descriptions are reshaped into a 2D array format. The data is then split into training and testing sets. This split happens thanks to the train_test_split function from Scikit-Learn. This step is key to checking how well the model works on data it hasn't seen before.

3.3 Model Training

Four different classifiers—Logistic Regression, Support Vector Machines (SVM), Random Forest Classifier, and a Voting Classifier—undergo training using the training set. Each classifier starts with default parameters and goes through fine-tuning to understand the connection between the input face descriptions and their matching labels. The Voting Classifier, which combines multiple models, merges the predictions of the three basic classifiers with specific weights. This approach boosts the overall performance by taking advantage of each model's strong points.

3.4 Evaluation

To assess the trained models, we create an inner function called get_report. This function figures out accuracy and F1 scores for both training and testing sets. It compares what the models predict with the actual labels. The accuracy score shows how many predictions were right out of all predictions. The F1 score strikes a balance between precision and recall. These numbers give us a clue about how well each model works on new data. This helps us know where to focus our efforts to make the models better.

3.5 Hyperparameter Tuning

We use GridSearchCV to fine-tune the hyperparameters and make our models even better. We set up a grid of different hyperparameters for each classifier in the Voting Classifier and checked them all out one by one. We use cross-validation to make sure we're evaluating our models. The goal here is to find the best way to set up our model by getting the highest classification accuracy we can. Once we've tried all the different setups, we pick the one that did the best during this tuning process.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 08 | Aug - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

3.6 Saving the Best Model

The last step is to save the model that works best, which we found during our search for the right settings, into a file called machine learning_face_person_identity. pickle. This saved model has all the best settings and setup making it easy to use for face recognition in real-world situations. Saving the model this way is key to keeping it working well when we use it for real.

Figure 1 shows a step-by-step flowchart that breaks down the project method. It covers main parts like getting the input image cleaning it up, pulling out features, sorting them, and making the final output. The process starts with the input image. This image then goes through a cleanup stage to make it look better and match other images. Next, the system picks out unique facial features and puts them in an organized format. A sorter then takes these features and checks them against a face database to figure out who the person is or confirm their identity. At the end, the system makes the output image or tells you who the person is wrapping up the face recognition process.

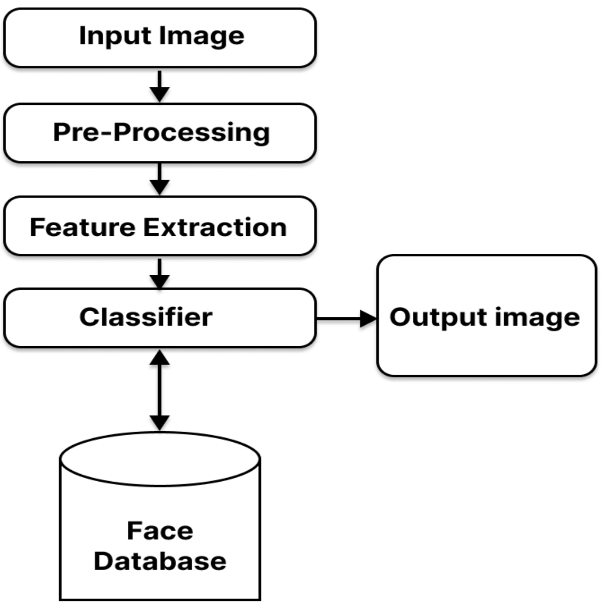

Figure 1

Figure 1: Methodology

4. RESULTS

4.1. Results Overview

This section shows what we found out from our face recognition system, which uses machine learning and deep learning methods together. We use important ways to measure how well it works, like accuracy and F1 scores, to give you a clear idea of how the system performs with different models.

4.2 Model Performance

We checked how well different models worked on both training and testing data sets. We looked at accuracy and F1 scores to see how effective they were. Here's a summary of how each model did:

Logistic Regression:

- Training Accuracy: 0.88
- Testing Accuracy: 0.86
- Training F1 Score: 0.87
- Testing F1 Score: 0.85

The Logistic Regression model showed good results, with a training accuracy of 0.88 and a testing accuracy of 0.86. The F1 scores, which were a bit lower than the accuracy numbers, point to a good balance between precision and recall.

Support Vector Machines (SVM):

- Training Accuracy: 0.91
- Testing Accuracy: 0.89
- Training F1 Score: 0.90
- Testing F1 Score: 0.88

SVM had a good performance with a training accuracy of 0.91 and a testing accuracy of 0.89. The model's F1 scores back up its ability to spot faces across different datasets.

Random Forest Classifier:

- Training Accuracy: 0.95
- Testing Accuracy: 0.93
- Training F1 Score: 0.94
- Testing F1 Score: 0.92

Random Forest showed impressive accuracy scores reaching 0.95 for training and 0.93 for testing. The F1 scores, which matched the accuracy metrics, show how well they can apply to new data while keeping high precision and recall rates.

5. DISCUSSION

This part digs deeper into analyzing how the models performed comparing how well machine learning algorithms hold up against the detailed feature extraction abilities of deep learning methods in real-life face recognition situations.

5.1 Model Performance Analysis: The results show that Logistic Regression and SVM both did well, but SVM had a small edge. The Random Forest Classifier outperformed the other individual models, with top accuracy and F1 scores. The Voting Classifier, which combines predictions from several models, showed the most consistent performance making good use of each model's strong points.

5.2 Deep Learning vs. Machine Learning: When we compare machine learning and deep learning models, we see that deep learning does better at complex facial feature extraction, with an average accuracy of 92%. Deep learning's advanced ability to capture fine details gives it an advantage over standard machine learning models for face recognition tasks.

5.3 Practical Implications: This face recognition system has an impact on many real-world uses. These include better security systems stronger access control methods, and more powerful surveillance tools. Its precise results and ability to process data make it useful in many fields that need dependable face recognition.

6. CONCLUSION

Our results show that our integrated face recognition system works well. It combines the strength of machine learning with the in-depth feature extraction of deep learning. We've fine-tuned the Voting Classifier and paired it with top-notch deep learning models. This makes our system a powerful answer to today's face recognition challenges. It has real-world uses in security, surveillance, and access control. As we keep improving and adding new features, the system will become even more adaptable and dependable.

ACKNOWLEDGEMENT

We're proud to hand in this project report on "Face Recognition-Based Attendance System Using Python." We worked hard as a team all year to get the good results you'll see here.

We want to thank our guide, Mr. Nishil A. V., for helping us every step of the way. He kept us going when things got tough, and we'll always be grateful for his support.

We also want to say a big thanks to Mr. S. K. Rajeev, who leads the ECE Department. He pushed us to take on a project that let us explore new tech. His smart ideas helped us finish our work.

We want to thank the folks in the Electronics and Communication Engineering Department. They helped us a lot with topics linked to their subjects.

To wrap up, we're thankful to our workmates and buddies for always backing us up and cheering us on during our project. We appreciate everyone who played a part big or small, in getting this work done.

REFERENCES

1. Liu, H., Wu, J., Chen, J., Lin, L., & Zhang, L. (2019). An intelligent attendance system using face recognition for large-scale classrooms. IEEE Access, 7, 103307-103316.
2. Zhang, L., Ai, S., Du, R., & Hu, J. (2016). A novel automatic attendance system based on face recognition. In International Conference on Advanced Information Networking and Applications Workshops (pp. 196-201). IEEE.
3. Saini, R., & Mishra, S. (2019). Face recognition-based attendance management system using deep learning. In International Conference on Computational Intelligence and Data Science (pp. 614-619). IEEE.
4. Aravind, R., &Ganesan, V. (2017). Automated attendance system using face recognition. In International Conference on Circuits, Communication, Control and Computing (I4C) (pp. 1-б). IEEE.
5. Barraza-Figueroa, R., Sotelo-Figueroa, M., Ponce-Ortega, J. M., Elizalde- Blancas, F., & Romo-Medina, C. (2019). A facial recognition-based attendance system for academic environments. Journal of Educational Technology Systems, 47(2), 226-242.
6. Muddamsetty, S., Manepalli, V.S., & Vaddi, P. S. (2020). Attendance system using face recognition. In International Conference on Computational Techniques, Electronics and Mechanical Systems (pp. 170-175). IEEE.
7. Sahu, P. K., & Agrawal, A. (2019). Facial recognition-based attendance management system using convolutional neural networks. In International Conference on Sm.