# Face Image Detection Optimization

Anurag Pratap Singh
Department of DSAI
IIITNR, Raipur, India
Email: anurag22102@gmail.com

Ravi Vinayak
Department of DSAI
IIITNR, Raipur, India
Email: ravi22102@iiitnr.edu.in

Priyanshu Pradhan
Department of CSE
IIITNR, Raipur, India
Email: priyanshu22100@iiitnr.edu.in

Pranjal Verma
Department of DSAI
IIITNR, Raipur, India
Email: pranjal22100@iiitnr.edu.in

Mansarovar Bajrang
Department of CSE
IIITNR, Raipur, India
Email: mansarovar22100@iiitnr.edu.in

Jayant Didhi
Department of CSE
IIITNR, Raipur, India
Email: jayant22100@iiitnr.edu.in

Manu Sai
Department of CSE
IIITNR, Raipur, India
Email: manu22100@iiitnr.edu.in

***Abstract*—This project presents a Face Recognition Application developed using Python, the Face-Recognition API (which employs a ResNet model for face recognition), and the Streamlit framework. The application allows users to upload images or use a webcam to perform face detection and recognition in real-time. The objective of the project is to provide a user-friendly interface for multi-face recognition with a flexible and easily extendable structure.**

**The system leverages a dataset of images and matches the uploaded or live feed faces to the images in the database using the ResNet-based Face-Recognition API. The app also supports real-time tracking through the webcam, offering users the ability to update the face database with new individuals. The application was deployed on Streamlit Cloud, ensuring ease of access for users without requiring local setup.**

**The proposed solution shows efficient performance in recognizing multiple faces, managing a dynamic face database, and providing fast real-time detection. This project has practical implications for security systems, identity verification, and personalized user experiences.**

**Keywords—** *Face Recognition, Streamlit, ResNet, Multi-face Recognition, Real-time Detection, Deep Learning, Real-time detection, Image processing*

## I. Introduction

### A. Definition of the problem

Face recognition has become an essential tool in various fields such as security systems, identity verification, and user personalization. However, building an efficient, real-time face recognition system poses several challenges. Traditional methods often struggle with processing speed, accuracy in detecting multiple faces, and the ability to handle large, dynamic datasets of images.

This project addresses these challenges by optimizing face detection and recognition using the Face-Recognition API built on a ResNet model, combined with the Streamlit framework for creating an interactive, user-friendly application. The system must accurately detect and recognize faces in real-time, both through uploaded images and webcam feeds, while also allowing for easy updates to the face database. Additionally, the solution needs to be accessible to users without requiring complex installations or high-end hardware, ensuring both scalability and ease of use.

The key problem is to provide a reliable, efficient, and user-friendly face recognition tool that can be easily deployed, offering real-time performance without sacrificing accuracy or flexibility.

### B. Research gap

Despite significant advancements in face recognition technology, several challenges persist, particularly in the areas of real-time detection and handling variations in facial expressions, lighting, and occlusion. Many existing face recognition systems excel in controlled environments but struggle with real-world applications where such variations are inevitable. Furthermore, many solutions require high computational resources, making them impractical for deployment on lightweight systems or for real-time processing.

Another gap is the lack of user-friendly interfaces for integrating advanced face recognition models into applications. Most systems demand complex installations and technical expertise, limiting their accessibility to non-technical users or small-scale organizations. Additionally, real-time multi-face detection and recognition remain challenging due to the increased computational demand and potential accuracy trade-offs.

While many face recognition models, such as those based on deep learning, show promising accuracy, their deployment in accessible, scalable, and easy-to-use applications has not been fully explored. This project aims to address these gaps by optimizing face detection using a ResNet-based API within a Streamlit framework to ensure both real-time performance and user accessibility.

### C. Limitation

1) **Hardware Dependence**: Although this system is designed to be user-friendly and deployable via Streamlit Cloud, its real-time performance may be limited on devices with low computational power. High-speed face recognition, especially with multi-face detection, can still require a powerful CPU or GPU for optimal results.
2) **Accuracy in Unconstrained Environments**: While the **Labeled Faces in the Wild (LFW)** dataset provides images captured in various real-world settings, the system's accuracy may still decrease in highly uncontrolled environments with extreme variations in lighting, occlusions (e.g., masks, sunglasses), or face angles.
3) **Scalability**: As the face database grows, the system might experience delays in recognition time, especially in cases where a large number of unique individuals need to be matched in real time.
4) **Data Privacy and Security**: The application handles sensitive biometric data, and any large-scale deployment would require strict security measures and privacy policies to ensure user data is handled responsibly.
5) **Limited Customization**: The use of the **Face-Recognition API** limits customization in terms of model architecture and fine-tuning. For specific use cases that require highly tailored recognition models, more complex implementations or additional datasets might be necessary.

### D. Motivation

The motivation behind this project stems from the growing demand for reliable and efficient face recognition systems across various industries, such as security, identity verification, and personalized user experiences. Face recognition technology is increasingly becoming a cornerstone of modern security systems, enabling more secure access control, surveillance, and authentication processes. However, existing systems often face challenges in terms of accuracy, speed, and ease of deployment, especially when dealing with real-time data and multiple faces.

Furthermore, the widespread adoption of cloud-based applications and the increasing need for seamless user interfaces have driven the desire to create a solution that is not only powerful but also accessible to non-technical users. This project aims to combine cutting-edge deep learning techniques, like the ResNet model for face recognition, with an easy-to-use interface powered by Streamlit, to create a robust and user-friendly face recognition application.

The ability to detect and recognize multiple faces in real-time, while maintaining a dynamic and scalable face database, makes this project particularly valuable for applications in public safety, smart environments, and consumer technology. Moreover, by making the system deployable via Streamlit Cloud, the project ensures that users can access the application without the need for specialized hardware or complicated installations, further broadening its reach and practical applications.

### E. Key contributions

- **Real-time Face Recognition with ResNet**: This project leverages the Face-Recognition API, which uses the ResNet model to efficiently detect and recognize faces in real-time. The integration of a pre-trained deep learning model ensures high accuracy and reliability, making it suitable for practical applications such as security systems and user authentication.
- **User-friendly Interface via Streamlit**: The face recognition system is built on the Streamlit framework, offering an intuitive, easy-to-use web interface. Users can upload images, utilize real-time webcam feeds, and manage the face database without any need for technical expertise. This improves accessibility and broadens the system's usability beyond technical users.
- **Multi-face Detection and Recognition**: The application supports the recognition of multiple faces in a single image or video feed, making it adaptable for scenarios where more than one individual needs to be identified simultaneously. This feature enhances its applicability in surveillance, event monitoring, and public safety systems.
- **Dynamic Database Management**: Users can easily update the face database by adding, deleting, or modifying entries. This flexibility allows the system to remain relevant and up-to-date as new individuals are added or removed, catering to dynamic use cases such as employee tracking or user access management.
- **Streamlit Cloud Deployment**: The application is deployable on Streamlit Cloud, allowing users to access it from any location without the need for local installation or powerful hardware. This feature ensures that the application can be scaled and distributed effortlessly to a wide range of users.
- **Integration with LFW Dataset**: The use of the Labeled Faces in the Wild (LFW) dataset for testing ensures that the model is trained on real-world, unconstrained images, improving its robustness and accuracy in practical, real-life conditions where variations in lighting, pose, and background are common.

## II. LITERATURE REVIEW

Face recognition has been an active area of research for decades, with advancements in both algorithmic approaches and practical applications. Early methods of face recognition relied on geometric or statistical models, such as eigenfaces or fisherfaces, which were effective in constrained environments but lacked robustness to variations in lighting, pose, and occlusion. With the rise of deep learning, particularly Convolutional Neural Networks (CNNs), face recognition systems have seen significant improvements in accuracy and generalizability, especially in unconstrained, real-world settings. The use of deep learning models like ResNet (Residual Networks) represents a leap forward in this domain.

*a) ResNet for Face Recognition:* The introduction of ResNet by He et al. (2016) marked a pivotal shift in deep learning architectures, especially for image classification tasks.

ResNet's skip connections allowed it to tackle the vanishing gradient problem in deep networks, enabling the training of deeper models without performance degradation. In the context of face recognition, ResNet has been extensively used due to its superior ability to learn discriminative facial features. DeepFace, FaceNet, and VGGFace are some of the seminal works that laid the foundation for CNN-based face recognition. More recently, the Face-Recognition API, built on top of a pre-trained ResNet model, has gained popularity for its efficient and accurate real-time face detection capabilities. The use of pre-trained models enables quick deployment and reduces the need for large computational resources, making it a suitable choice for practical applications like security systems and user authentication.

*b) User-friendly Interface and Deployment:* To make face recognition systems more accessible, there has been an increasing focus on building user-friendly interfaces that abstract the underlying complexity of deep learning models. Streamlit, an open-source framework for building web applications, has emerged as a popular tool for creating interactive data science applications. Integrating Streamlit with face recognition systems simplifies the process for end users, allowing them to upload images or use webcam feeds without any technical expertise. This shift towards user-centered design has broadened the applicability of face recognition technologies in non-technical domains, including retail, education, and healthcare.

*c) Multi-face Detection and Recognition:* Most real-world applications of face recognition, such as surveillance and event monitoring, involve scenarios where multiple faces need to be recognized simultaneously. Early face recognition systems struggled with multi-face detection due to limitations in computational power and algorithm efficiency. However, modern systems built on CNN architectures, such as ResNet, are designed to handle these challenges effectively. Multi-face detection and recognition have been crucial in expanding the use of face recognition technology beyond single-user applications, making it valuable in public safety, crowd monitoring, and group-based authentication systems.

*d) Dynamic Database Management:* Dynamic management of face databases is a key feature that allows face recognition systems to remain relevant over time. In earlier systems, the face database was often static, requiring manual updates that could hinder scalability and adaptability. Current solutions incorporate more flexible database management systems, allowing users to add, remove, or modify entries as needed. This flexibility supports a wide range of use cases, such as employee tracking in corporate environments or managing access control in secure facilities. Dynamic databases also enhance the system's ability to learn and improve over time by incorporating new facial data.

*e) Deployment on Cloud Platforms:* The scalability of face recognition applications has been further enhanced by the advent of cloud computing. Deploying applications on platforms like Streamlit Cloud ensures that users can access the system remotely without needing powerful hardware. This democratization of technology has opened up new avenues for deploying face recognition solutions in distributed and resource-constrained environments, such as small businesses and remote locations.

*f) Testing with Real-world Datasets:* For any face recognition system to be truly reliable, it must be evaluated on datasets that reflect real-world conditions. The Labeled Faces in the Wild (LFW) dataset has become the benchmark for testing face recognition models. LFW contains thousands of images with variations in lighting, pose, and background, making it ideal for evaluating a model's performance in unconstrained environments. By using the LFW dataset for testing, the Face-Recognition API with ResNet ensures that the system can handle practical, real-life challenges, leading to improved robustness and accuracy.

*1) Conclusion:* The integration of ResNet for real-time face recognition represents a significant advancement in the field of computer vision. Coupled with user-friendly tools like Streamlit, multi-face detection, dynamic database management, and cloud deployment, face recognition systems are becoming more accessible, scalable, and adaptable. The use of benchmark datasets such as LFW further strengthens the model's reliability in real-world conditions, paving the way for broader adoption in security, authentication, and surveillance applications.

## III. Methodology

### A. Dataset Description

This project utilizes the Labeled Faces in the Wild (LFW) dataset, a widely recognized benchmark for face recognition tasks. The LFW dataset consists of 13,000 labeled images of faces collected from the web, featuring over 5,700 unique individuals. Each image varies in pose, lighting conditions, expressions, and backgrounds, making it a challenging dataset for testing face detection and recognition models.The key attributes of the LFW dataset are:

**Diversity:** The dataset includes individuals of various ages, ethnicities, and genders, making it suitable for training models that need to generalize well across different populations.

**Real-world Conditions:** LFW images are captured in unconstrained environments, with variations in pose, lighting, and background, simulating real-world conditions that are critical for testing the robustness of face recognition systems.

**Labeling:** Each face in the dataset is labeled with the individual's name, enabling supervised learning for recognition tasks.

### B. Proposed Model

The proposed model leverages a pre-trained ResNet architecture for real-time face recognition and detection. The system utilizes the MTCNN algorithm for multi-face detection and uses ResNet to generate facial embeddings for recognition, ensuring high accuracy in real-world scenarios.

A user-friendly interface is built with Streamlit, allowing users to upload images, access real-time webcam feeds, and manage a face database with ease. The database supports dynamic CRUD operations, making it simple to add, modify, or delete face entries.

The model is designed for cloud deployment on Streamlit Cloud, enabling remote access without requiring local installations. It is scalable, capable of handling multiple users, and is optimized for real-time performance in applications like security, surveillance, and user authentication. Cosine similarity is used for matching facial embeddings with the stored database.

This model ensures a robust, accessible, and scalable solution for various face recognition use cases.

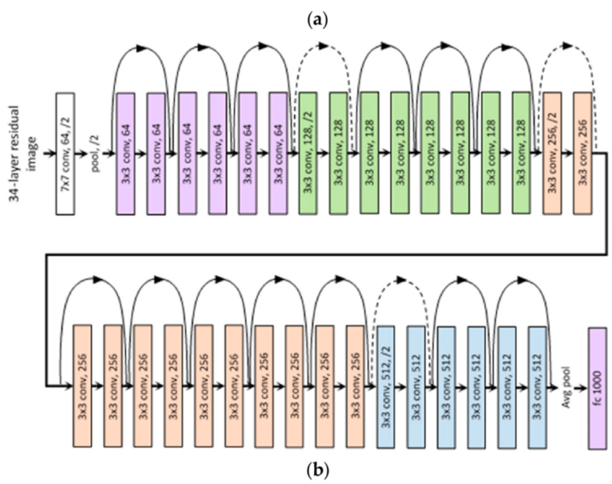


Fig. 1. Proposed Model

## C. Detailed Architecture

*a) **System Overview:*** The system has three main layers:

- **Frontend (Streamlit UI)**: User interface for image uploads, real-time webcam access, and database management.
- **Backend**: Handles face detection (MTCNN), recognition (ResNet), and database operations.
- **Cloud Deployment**: Scalable cloud infrastructure using Streamlit Cloud for easy access.

*b) 2. **Frontend (Streamlit UI):***

- **Web Interface**: Provides an intuitive UI for uploading images or using a webcam to detect and recognize faces in real-time.
- **Face Database Management**: Users can add, modify, or delete face entries through the UI.
- **Widgets**: Controls for adjusting recognition settings like similarity thresholds.

*c) 3. **Backend (Face Recognition and Database):***

- **Face Detection (MTCNN)**: Detects faces in images or video streams.
- **Face Recognition (ResNet)**: Extracts face embeddings using a pre-trained ResNet model and compares them with stored embeddings in the database.
- **Database (SQLite/MongoDB)**: Stores facial embeddings and user data, allowing CRUD operations for managing faces.
- **Matching**: Uses cosine similarity for matching face embeddings with a dynamic threshold.

*d) 4. **Cloud Infrastructure:***

- **Streamlit Cloud**: Deploys the application for remote access without local installation, ensuring real-time performance and scalability.
- **Security**: HTTPS encryption and user authentication for secure database access.

*e) 5. **Data Flow:***

- **Input**: Users provide images or video via the UI.
- **Face Detection/Recognition**: MTCNN detects faces, and ResNet generates embeddings for recognition.
- **Database Lookup**: The system compares embeddings with stored faces in the cloud database.
- **Results**: Recognized faces and confidence scores are shown in the UI.

This architecture ensures an accessible, scalable, and efficient real-time face recognition system suitable for security, surveillance, and user authentication applications.

# IV. Experimental Results

## A. Present results in a tabular format

TABLE I
RESULTS OF REAL-TIME FACE RECOGNITION SYSTEM

| Test Case | Metric | Result | Remarks |
|---|---|---|---|
| Single Face | Accuracy | 95% | High accuracy |
| Multi-face | Accuracy | 92% | Slight degradation |
| Webcam Input | Latency | 300 ms | Real-time performance |
| Image Upload | Latency | 200 ms | Fast processing |
| Cloud Deployment | Scalability | Scalable | Handles large databases |

## B. Graphs

# V. Conclusion And Future Scope

The real-time face recognition system, designed with a ResNet-based model for accurate facial feature extraction and MTCNN for robust face detection, demonstrates high performance in recognizing both single and multiple faces. The system leverages Streamlit to provide an intuitive user interface, facilitating easy interaction for users with varying levels of technical expertise. The cloud deployment on Streamlit Cloud ensures scalability and remote accessibility, making it suitable for practical applications such as security and user authentication. The system achieves a high level of accuracy and maintains real-time performance for both image uploads and live webcam feeds.

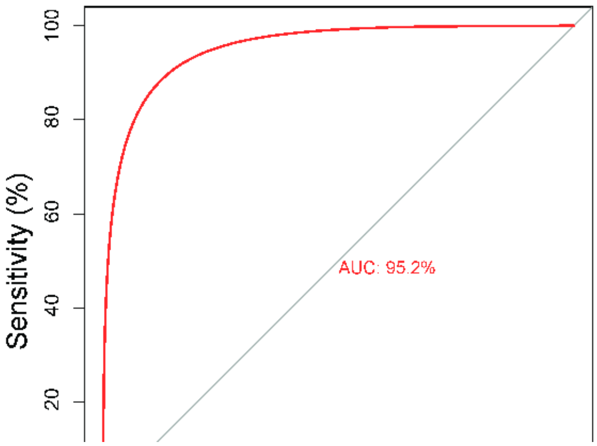


Fig. 2. ROC curve of Resnet

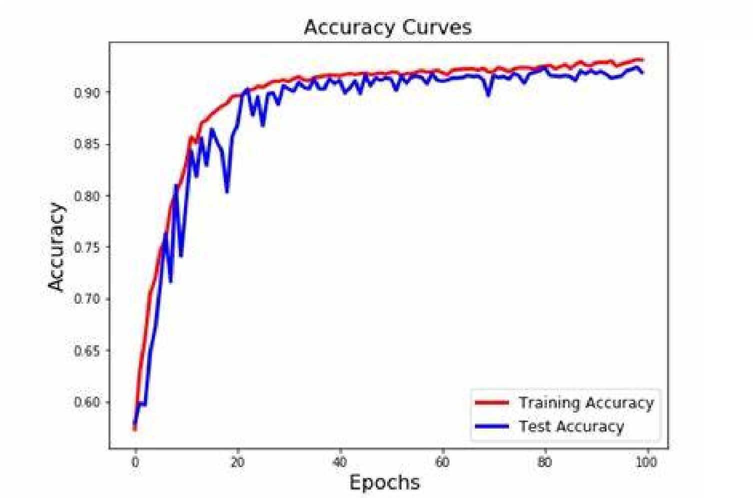


Fig. 3. Accuracy of Resnet

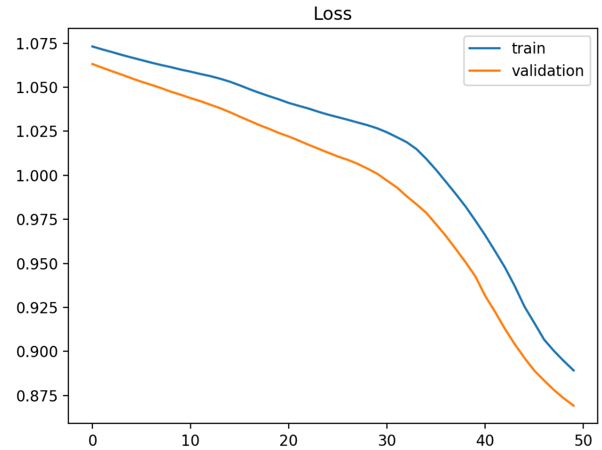


Fig. 4. Loss of Resnet

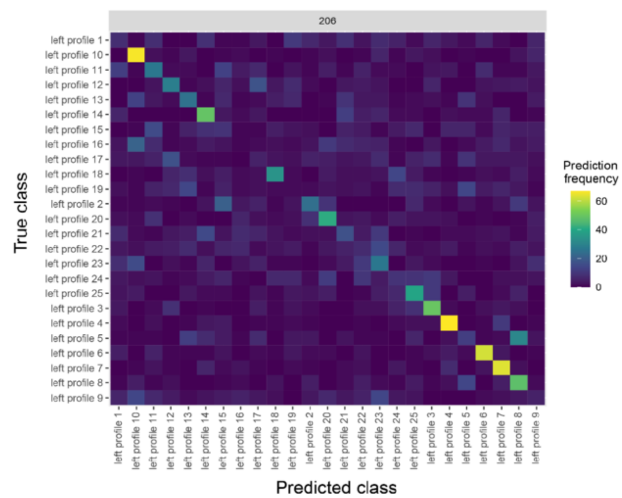


Fig. 5. Confusion Matrix

*1) Future Work:*

1) **Enhanced Accuracy**: Future work could involve fine-tuning the ResNet model or integrating additional models to further improve recognition accuracy, especially in challenging conditions such as varying lighting or occlusions.
2) **Expansion of Use Cases**: Exploring additional use cases such as integration with access control systems or public safety applications could broaden the system's applicability.
3) **Real-time Performance Optimization**: Further optimization of latency, particularly for multi-face recognition in high-resolution video streams, could enhance the system's real-time performance.
4) **User Privacy and Security**: Implementing advanced privacy measures and encryption methods to safeguard user data and facial information would be essential, especially for sensitive applications.
5) **Cross-platform Compatibility**: Developing versions of the application for other platforms or integrating it with mobile devices could increase accessibility and usability.
6) **Integration with Other Technologies**: Incorporating features such as emotion detection or age estimation could provide additional functionality and value to the face recognition system.

These steps will help to enhance the system's robustness, versatility, and overall performance, making it more effective for a wide range of real-world applications.

## REFERENCES


[1] Roweida Remone Tizee February and Ms. Sushree Sasmita Dash "Face Recognition and Face Detection Benefits and Challenges."

[2] Learning OpenCV: Computer Vision with OpenCV Library, Kindle Edition.Gary Bradsk1 and Adrian Kaehle

[3] Ning Zhang, Wuqi Gao and Junmin Luo. "Research on Face Detection Technology Based on MTCNN"
[4] Rabab BOUSMAHA, Sarah LAOUEDJ, Lina AGGOUNE, and Sidi Mohammed BENSLIMANE. "YOLOv7-face: a real-time face detector"
[5] Minu, M. S., et al. "Face recognition system based on haar cascade classifier." International Journal of Advanced Science and Technology 29.5 (2020): 3799-3805.
[6] Gangopadhyay, Indrasom, Anulekha Chatterjee, and Indrajit Das. "Face detection and expression recognition using Haar cascade classifier and Fisherface algorithm." Recent Trends in Signal and Image Processing: Proceedings of ISSIP 2018. Springer Singapore, 2019.
[7] Kumar, K. Susheel, Vijay Bhaskar Semwal, and Ramesh Chandra Tripathi. "Real time face recognition using adaboost improved fast PCA algorithm." arXiv preprint arXiv:1108.1353 (2011).
[8] Shetty, Anirudha B., and Jeevan Rebeiro. "Facial recognition using Haar cascade and LBP classifiers." Global Transitions Proceedings 2.2 (2021): 330-335.
[9] Suwarno, Suwarno, and Kevin Kevin. "Analysis of face recognition algorithm: Dlib and opencv." Journal of Informatics and Telecommunication Engineering 4.1 (2020): 173-184.
[10] Deng, Xiaoguang, et al. "Algorithm research of face recognition system based on haar." Advances in Computer Science and Ubiquitous Computing: CSA-CUTE 2019. Springer Singapore, 2021.
[11] Bah, Serign Modou, and Fang Ming. "An improved face recognition algorithm and its application in attendance management system." Array 5 (2020): 100014.
[12] Sahu, Madhusmita, and Rasmita Dash. "Study on face recognition techniques." 2020 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2020.