Face Identification and Categorization with and without Mask utilizing MTCNN and OpenCV for accessing Bank Locker Facility

Ms. R.Monica Lakshmi[1], Nikita Devendran[2], Deepika L.P[3], Mounika.S[4],

[1] Assistant Professor, Department of Computer Science and Business Systems, R.M.D. Engineering College, Tamil Nadu - 601 206, monica.ramar@gmail.com

[2] Student, Department of Computer Science and Business Systems, R.M.D. Engineering College, Tamil Nadu - 601 206, ucb20128@rmd.ac.in

[3] Student, Department of Computer Science and Business Systems, R.M.D. Engineering College, Tamil Nadu - 601 206, ucb20111@rmd.ac.in

[4] Student, Department of Computer Science and Business Systems, R.M.D. Engineering College, Tamil Nadu - 601 206, ucb20126@rmd.ac.in

Abstract - Face recognition is an application-rich, widely-used technology. However, accurate facial identification is challenged by the emergence of face masks, especially during the COVID-19 pandemic. The goal of this research is to create a facial recognition system that can identify people even while they are wearing masks. They will be able to open the locker once the face has been confirmed. If not, entry would be refused. First, faces in the stream are identified using face identification techniques such as MTCNN. Next, each recognized face's orientation and scale are normalized through the use of an alignment procedure. Feature extraction using a deep learning model, like Facenet, is the next stage. With the help of this model, every aligned face is converted into an embedding—a high-dimensional vector representation that captures the key traits and features of the face. Ultimately, the embedded vectors are subjected to a softmax classifier in order to authenticate or classify faces. Furthermore, activities like face verification and grouping can be made possible by using similarity measures like cosine similarity to evaluate the similarity of embedded vectors.

Keywords : MTCNN, Facenet, OpenCV, Classification, Face Alignment, Face Recognition.

I. INTRODUCTION

In contemporary security systems, facial recognition and classification are essential components, particularly in settings like bank locker access. However, conventional facial recognition techniques are facing serious hurdles as a result of the increased acceptance of mask wear for a variety of reasons, including health concerns. Novel techniques are being investigated to tackle this problem, utilizing tools such as OpenCV and the Multi-task Cascaded Convolutional Neural Network (MTCNN). These technologies make it possible to create systems that can correctly recognize and classify people, masks or not. The project's goal is to create a facial recognition system that can reliably identify people even while they are mask-wearing. The system correctly links the recognized face to its matching identity in the database by using extensive matching algorithms based on deep learning. By providing dependable access control to bank locker facilities, this solution not only strengthens security measures but also highlights how flexible facial recognition systems are in constantly shifting contexts. Furthermore, these solutions take a forward-thinking approach to addressing modern difficulties by supporting both masked and unmasked scenarios, all the while upholding the integrity and effectiveness of the required security procedures.

II. LITERATURE SURVEY

[1] Rahul Baghel, Pallavi Pahadiya, Upendra Singh, *"Human Face Mask Identification using Deep Learning with OpenCV Techniques"*, 7th International Conference on Communication and Electronics Systems (ICCES), 2022, pp. 1051-1057.

The COVID 19 virus is transmitted by respiratory droplets and interpersonal contact, according to the WHO. It is advised to wear masks and isolate oneself from others in order to stop the spread. A mask recognition system within video streams has been created to meet regulatory standards. A deep learning method using the Python TensorFlow, Keras, and PyTorch packages are used to detect masks.

[2] Amjad Ali Khan, Shankar Yadav, Lalit Kumar, "Face Mask Detection Using OpenCV and Machine Learning", 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2022, pp. 80-85.

Protective face mask wear has emerged as a new fashion and is now accepted. As a result, we are creating an AI system that can detect whether or not a person is wearing a mask. We can stop the virus from spreading and safeguard the ecosystem with the aid of this method. This project requires the installation of numpy, OpenCV, TensorFlow, and other learning tools in addition to Jupyter Notebook.

[3] Tamilarasan Kumar, Rajendrane Rajmohan, Muthu Pavithra, "Automatic Face Mask Detection System in Public Transportation in Smart Cities Using IoT and Deep Learning", Expert Systems with Applications,Volume 36, Issue 2, Part 2.

In order to lessen the effects of COVID-19, there is an increased demand for effective city management due to population growth. AI should be used to expand smart cities if they are to thrive. An IOT-based face mask detection system for public transit is presented in this article. Real-time data is collected by this system. The model can detect faces and masks with minimal inference time and memory, which satisfies the constrained resources according to the trials.

[4] Gopinath Pranav Bhargav, Kancharla Shridhar Reddy, Alekhya Viswanath, "Facemask Detection with Face Recognition and Alert System using MobileNetV2", 2nd International Conference on Intelligent and Cloud Computing (ICICC), 2022, vol. 286, pp. 77-87.

Donning a mask is a successful strategy for preventing COVID-19. We developed a web application that uses an integrated facemask detection and face-recognition algorithm to continuously remind users to wear masks. The suggested method recognizes a person's face and determines if they are wearing a mask or not. Additionally, an automatically produced email is sent to that particular violator's personal email address reminding him to wear a mask.

[5] K. Yagna Sai Surya, T. Geetha Rani, B. K. Tripathy, "Social Distance Monitoring and Face Mask Detection Using Deep Learning", Computational Intelligence in Data Mining, 2022, pp.461-476.

Deep learning frameworks and Python can be used to create a reliable system for tracking social distancing and people capture. The technology can reliably determine whether people are wearing masks by utilizing deep learning models, which guarantees adherence to safety protocols. By utilizing computer vision techniques, the system may effectively monitor social distance measures by continuously observing persons in real-time, regardless of the time of day. In public areas, shopping malls, and other settings where keeping safe distances is essential to halting the spread of illness, this technology is quite helpful.

III. EXISTING SYSTEM

In the state of face recognition systems today, facial feature analysis is the primary method of identification. Nevertheless, these systems have significant difficulties in correctly detecting people who are donning masks. Conventional facial recognition algorithms rely largely on the capture of unique facial features such as the mouth, nose, and chin, which are frequently covered up, either completely or in part, by masks. As such, the performance of current facial recognition algorithms noticeably deteriorates in situations where mask use is common.

This deterioration in performance manifests through various significant disadvantages, including diminished accuracy in identification, limitations in detecting masked faces, increased computational complexity due to the need for additional preprocessing steps or alternative feature extraction methods, and heightened privacy concerns stemming from potential intrusions into individuals' privacy when attempting to recognize faces obscured by masks. Addressing these challenges requires the development of innovative approaches that can handle facial recognition.

IV. PROPOSED SYSTEM

Developers can use face recognition algorithms to reliably identify human faces from a stream against a database of recognized faces by utilizing Python and its associated modules.

First, faces in the stream are identified and located using face detection techniques such as MTCNN. After the faces are identified, an alignment procedure is utilized to standardize the orientation and scale of every identified face, guaranteeing uniform input for the following phases.

After face detection and alignment, a deep learning model like Facenet is used for feature extraction. With the help of this model, every aligned face is converted into an embedding—a high-dimensional vector representation that captures the key traits and features of the face. These embeddings provide effective comparison and recognition by acting as compact and discriminative representations of faces.

Ultimately, the embedded vectors are subjected to a softmax classifier in order to authenticate or classify faces. Furthermore, activities like face verification and grouping can be made possible by using similarity measures like cosine similarity to evaluate the similarity of embedded vectors.

V. METHODOLOGY OF APPROACH

A. System Specifications

The software requirements are:

- Anaconda Distributor.
- Python - 3.7 or later versions.
- Text editor such as Notepad, or MS Word.
- A 64-bit operating system with a x64-based processor.
- Windows 7 or later versions.
- Visual Studio Code.

The hardware requirements are:

- Core i3 processor and above.
- 4 GB of RAM.
- 80 GB of hard disk capacity.
- 15 inch color monitor.
- Full HD webcam.
- 52x CD-ROM drive.

B. Architecture Diagram

Architecture diagram is a visual representation of software system components. The below diagram is the architecture of the system.

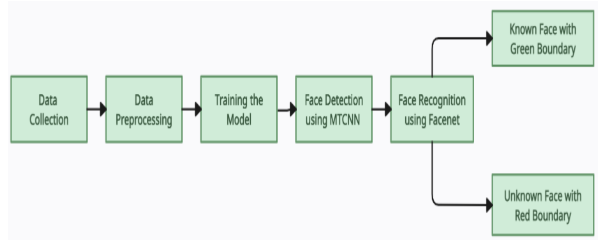

Fig.5.1 Architecture diagram

The process of collecting data entails compiling an array of distinct facial picture datasets of each person, containing a range of positions, expressions, and lighting scenarios captured from the webcam's real-time feed.

Next, in order to increase model generalization, the data preprocessing module standardizes and improves the gathered images by resizing, normalizing, and enhancing them.

During the training phase, the preprocessed data is fed into a recognition model (e.g., Facenet) so that it may learn to produce embeddings and extract discriminative features from faces.

MTCNN is used for face detection, precisely locating and extracting facial regions from pictures or video frames. After being identified, these faces are aligned and sent into Facenet so that embeddings may be extracted.

Lastly, the trained model is applied to real-time face recognition tasks, where people are identified or verified by comparing embedded vectors using methods such as cosine similarity.

C. Libraries and Frameworks

The libraries and frameworks used in this system are:

- Matplotlib.
- Keras.
- TQDM.
- Scipy.
- Pickle.
- Utils
- Glob
- Argparse
- Numpy
- CV2

D. Data Collection

Data collection is a crucial step in developing a face recognition system. The process involves gathering relevant data to train the face detection, and face recognition algorithms. Collect a diverse dataset of face images representing different individuals. These images should cover variations in age, gender, ethnicity, and facial expressions. The webcam is used to take the pictures. To get realistic photos, place the subject squarely in front of the camera under well-lit conditions. To begin the collection process, a virtual environment needs to be built using the Anaconda prompt. Activate the environment once it has been built. As a consequence, the system gathers about 100 photographs. When a person is identified, their name will be presented in the directory under their respective name.

E. Data Preprocessing

Data preprocessing refers to the process of transforming raw data into a more suitable for analysis or machine learning tasks.

Generating fresh versions of the training images that represent the scenarios the model faces during inference is the aim of image data augmentation. The model gains the ability to handle various conditions by introducing these changes. It is frequently used in conjunction with Python libraries to add them to the pipeline for training. Rotation, translation, scaling, flipping, shearing, zooming, brightness and contrast alteration, and noise injection are some of the frequently utilized methods in image data augmentation. These methods can be applied separately or in combination, based on the needs.

F. Training the Model

The act of utilizing a machine learning algorithm to identify patterns and relationships within a dataset is known as "training the data." A machine learning model is trained with a sizable collection of tagged face photos in the context of facial recognition. The model gains the ability to identify pertinent features and transfer them to the appropriate labels during training.

"Training the data" refers to the process of using a machine learning algorithm to find patterns and relationships within a dataset. For facial recognition, a large set of annotated face pictures is used to train a machine learning model. During training, the model learns to recognize relevant features and map them to the corresponding labels.

After alignment, the facial images are run through Facenet, an embeddings model. A face that has been taken from an image and represented numerically is referred to as embedding. A face recognition system converts an image into a compact numerical high-dimensional vector representation, or embedding, by using a deep learning model to extract features from the face.

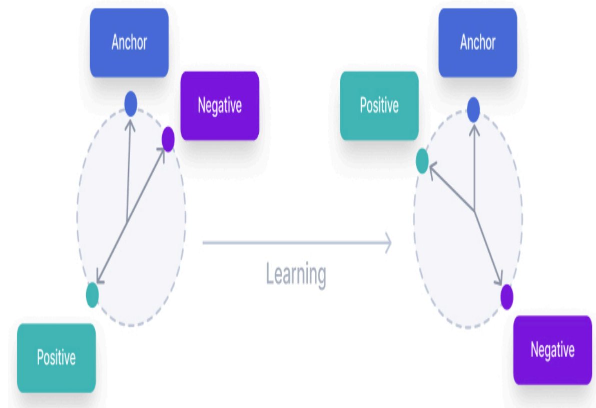

Fig.5.2 Triplet Loss in Facenet

Similar faces are mapped in this space close to one another, whereas dissimilar faces are mapped farther apart, according to the way this embedding is learned. FaceNet's use of a triplet loss function during training is one of its main advances. A negative picture (representing a different identification), a positive image (representing the same identity as the anchor), and an anchor image (representing the identity to be recognized) are the three images whose similarity is compared by this loss function.

The goal of the model's training is to maximize the distance between the anchor's embeddings and negative images while minimizing the distance between the anchor's embeddings and positive images. This promotes the model's learning of discriminative embeddings that are also invariant to lighting, posture, and facial emotions.

G. Face Detection

Face detection is the task of detecting a human face on an image. This is done through extracting a list of bounding boxes, coordinates of smallest possible rectangles around faces.

We use MTCNN to perform face detection. MTCNN leverages a 3-stage neural network detector. First, the image is resized multiple times to detect faces of different sizes. Then the P-network (Proposal) scans images, performing first detection. The proposed regions are input for the second network, the R-network (Refine), which filters detections to obtain quite precise bounding boxes. The final stage of MTCNN is the O-network (Output) which performs the final refinement of all the bounding boxes.

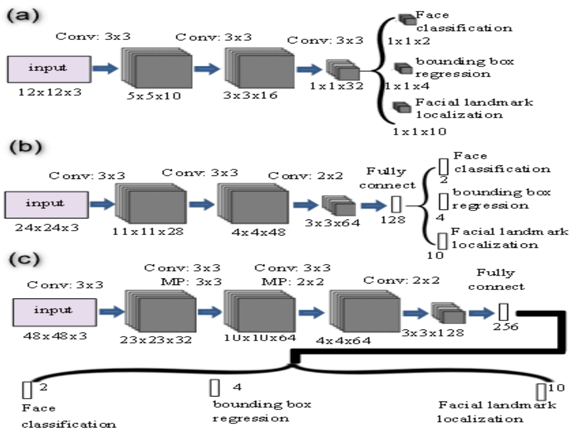

Fig.5.3 MTCNN Overview

H. Face Alignment

The process of identifying and locating important facial landmarks on a face, followed by aligning or posing the image to a standard is known as face alignment. Typically, these landmarks consist of points like the corners of the mouth, nose tip, and eyes.

Face alignment is used to make sure that, despite changes in head position, facial expression, or illumination, facial features are positioned and aligned consistently in various photographs. Facial recognition algorithms can compare and match facial features more precisely for identification or verification when faces are aligned to a shared reference frame.

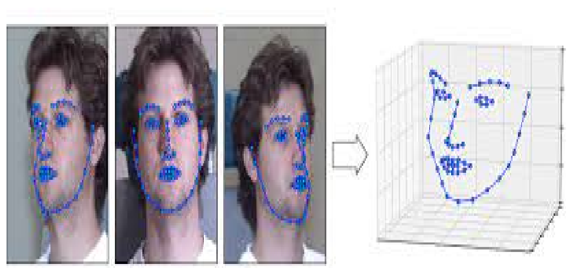

Fig.5.4 Face Alignment

I. Classification

A classifier is a machine that determines if a picture is positive or negative. To accurately identify a new image as a face or non-facial image, it is trained on hundreds of thousands of face and non-face photos. To find things, we employ the Haar Classifier, which instills a cascade operation from the images. Next, the traits are extracted from the image. Every characteristic is a single number that is obtained by deducting the total of the pixels in the model's predicted output from the actual target values in the training dataset. It represents the discrepancy between the predicted and actual values and is used as a measure of how well the model is performing.

The Haar Value Classification is as follows:

$$\text{Pixel value} = (\text{Sum of the Dark pixels}/\text{Number of Dark pixels}) - (\text{Sum of the Light pixels}/\text{Number of Light pixels})$$

No.of faces	Execution Time (sec)	No.of faces detected	Accuracy (%)
5	0.141	5	100
10	0.055	9	90
15	0.11	12	80
20	0.369	19	95

Table 5.1 Observation of Haar

The percentage of accurate predictions the model made on the training dataset is known as accuracy. The difference between the model's predicted output and the actual target values found in the training dataset is measured by loss.

To turn raw scores or logits into probabilities, a neural network's output layer applies the softmax function. The possibility that each class or identity will be present in the input data is represented by these probabilities. It receives as input facial image features and generates a vector of scores representing various identities or classifications.

$$P(class_i) = \frac{e^{score_i}}{\sum_j e^{score_j}}$$

where, P(class i) is the probability of the i-th class, score i is the raw score or logit for the i-th class, e is the base of the natural logarithm, and the summation j is the sum of the exponentials of all raw scores.

If two vectors are roughly pointing in the same direction, their cosine similarity can be calculated using the cosine of their angle. Cosine similarity measures the similarity between two faces by translating them into these numerical representations.

Perfect similarity is represented by a value of 1, total dissimilarity is represented by a value of -1, while values closer to 0 indicate a lack of meaningful resemblance. choosing the match in face identification to be the one with the highest cosine similarity score.

The cosine similarity between the two feature vectors A and B, can be calculated by the following formula :

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| \, ||\mathbf{B}||} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

An epoch refers to one complete pass of the entire training dataset through the neural network during the training phase.

In order to reduce the error between the anticipated outputs and the actual labels, the neural network uses optimization algorithms like gradient descent to modify its parameters (weights and biases) during training. Iteratively going through several epochs, this procedure is carried out until the model converges to a performance level that is suitable. The training dataset is split up into smaller batches throughout each epoch, which comprises several iterations. The batches are successively fed into the neural network, and the model's parameters are adjusted based on the accumulated gradients once all of the batches have been processed in an epoch.

The accuracy of a machine learning model during testing and training is displayed as the number of epochs increases on the graph. "Epochs" is the label on the x-axis of the graph, while "accuracy" is the label on the y-axis. The model's performance on the data it was trained on is indicated by the training accuracy, while its performance on data that hasn't been seen yet is indicated by the testing accuracy. The training accuracy in the graph indicates that the model is perfectly fitting the training data because it starts at 1.0 and remains there throughout all epochs. When the number of epochs rises, the testing accuracy varies slightly from its initial value of 0.98.

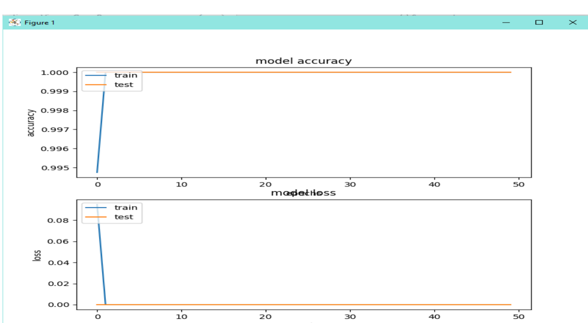

Fig.5.5 Epoch Calculation

J. Face Identification

A high-dimensional vector space is created by encoding faces into FaceNet, a facial recognition framework, so that similar faces are situated next to one another.

FaceNet recognizes faces in images and uses preprocessing to prepare the image before feeding it into the neural network to create an embedding, or numerical representation, of the face. The similarity between the faces is then ascertained by comparing these embeddings using cosine similarity or alternative distance metrics

Finding and locating faces in each video stream frame is the initial step in the process. Convolutional neural networks and other deep learning-based approaches, such as Haar cascades, are used for this. For improved recognition accuracy, faces may need to be aligned and normalized to a standard size and orientation after they have been discovered. Facial features are taken from the aligned faces after normalizing. In this technique, significant facial features are recorded, such as the locations of significant landmarks or numerical depictions of facial patterns. Finally, similarity metrics like cosine similarity and Euclidean distance are used to compare the retrieved characteristics to a database of recognized faces or templates. If more than a particular number of matches are identified, the individual's

VI. RESULT AND DISCUSSION

Accurate identification and verification while people wear masks is a problem that is addressed by the face recognition system that uses MTCNN and OpenCV both with and without mask detection. The technology increases overall accuracy and dependability in identifying people, even while they are wearing masks, by merging facial recognition algorithms with mask detection. Additionally, it prevents faces that aren't educated from using the locker system. It is therefore superior to the conventional mask recognition system.

A.Elapsed Duration

It refers to the amount of time it takes for the system to process and analyze a given image or video frame to identify faces. When considering the presence of masks, traditional systems struggle to accurately identify individuals, requiring additional steps such as manual verification. In contrast, the new facial recognition systems have adapted to accommodate mask-wearing, effectively reducing the time.

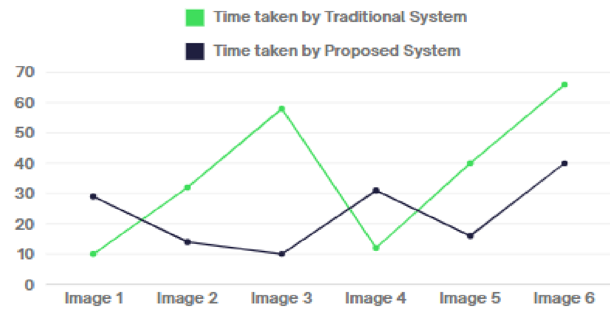

Fig.6.1 Elapsed Duration

B.Accuracy

Accuracy refers to the ability of the system to correctly identify or verify individuals from their facial features. It is typically measured as the percentage of correct identifications out of the total number of attempts. Without masks, both systems exhibit comparable accuracy, but the new system demonstrates its effectiveness in the presence of facial coverings.

Fig.6.2 Accuracy

C. F1 Score

It is a metric used to evaluate the performance of a facial recognition system. F1 score is the harmonic mean of precision (proportion of true positives among all instances classified as positive) and recall (proportion of true positives among all actual positive instances). The score decreases significantly when individuals wear masks due to occlusion of facial features. With all-new facial recognition systems, the score remains robust even in masked scenarios.

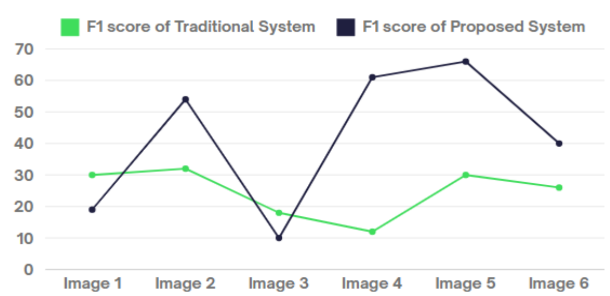

Fig.6.3 F1 Score

D. Resource Consumption

Resource consumption refers to the amount of computational resources required for the system to operate effectively. Traditional facial recognition systems rely on processing the entire facial structure, consuming considerable computational resources. The proposed system with mask detection algorithms can process facial features even when partially covered, thereby reducing computational overhead.

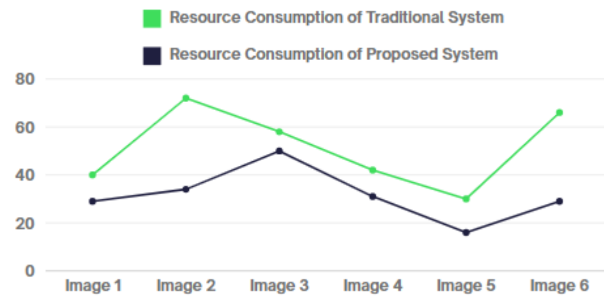

Fig.6.4 Resource Consumption

E. Environmental Resilience Capacity

It involves the system's capacity to adapt and function effectively despite the environmental variables. Traditional systems struggle in environments where factors like lighting changes, partial obstructions, are present making them less resilient. The proposed system, especially designed to adapt to these challenges, demonstrates superior environmental resilience.

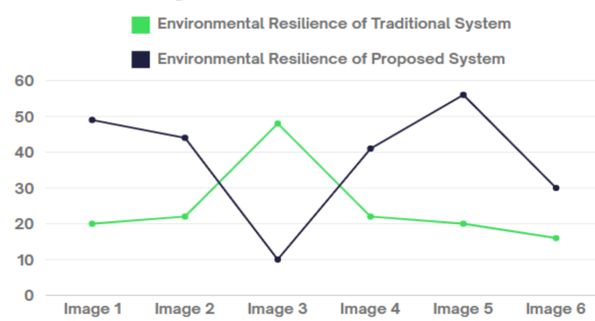

Fig.6.5 Environmental Resilience

F. Cross Domain Performance

Cross-domain performance refers to the assessment of a system's effectiveness when deployed across various domains or applications. This evaluation is crucial for understanding how well the system performs in different contexts and scenarios, such as surveillance, access control, or user authentication.This involves assessing its ability to achieve consistent and reliable results across diverse environments, datasets, and plenty of real time situational use cases.

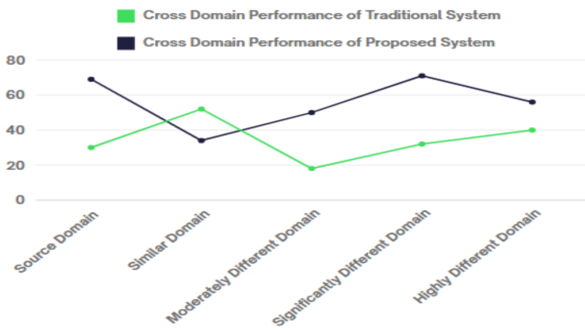

Fig.6.6 Cross Domain Performance

VII. FUTURE ENHANCEMENTS

- Improved Mask Detection: Enhance the accuracy and robustness of the mask detection algorithm by incorporating advanced machine learning techniques, such as deep learning models or ensemble methods.
- Multi-Modal Fusion: Investigate the integration of multiple biometric modalities, such as facial features, voice recognition, gait analysis, or even thermal imaging, to further enhance the identification accuracy.
- Integration with Bank Locker Access System : Develop APIs or protocols for communication and integration between the face recognition module and the bank's locker access system, ensuring the real-time authentication and authorization of individuals.
- Real-time Monitoring and Logging: Implement real-time monitoring of individuals entering the bank locker facility, capturing their faces and mask status. Log this information along with timestamps for auditing purposes and to ensure compliance with security protocols.
- Scalability and Robustness: Ensure that the system is scalable to handle a large number of users and robust enough to perform reliably in real-world conditions.

VIII. CONCLUSION

The face recognition system integrates MTCNN and OpenCV to improve identification accuracy even with masks. By combining mask detection with face recognition, it enhances security and access control, ensuring reliability across different environments. Incorporating alternative biometric modalities supplements identification in mask-wearing scenarios, prioritizing privacy by focusing on identification only when masks are absent. In summary, this system offers accurate identification, enhanced security, and adaptability to mask-related challenges.

REFERENCES

[01] G. Singh, I. Gupta, J. Singh and N. Kaur, "*Face Recognition using Open Source Computer Vision Library with Python*," 10th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO), Noida, India, 2022, pp. 1-6.

[02] A. Kumari Sirivarshitha, K. Sravani, K. S. Priya and V. Bhavani, "*An approach for Face Detection and Face Recognition using OpenCV and Face Recognition Libraries in Python*," 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 1274-1278.

[03] J. Zhang, "*Online Face Recognition System Based on Convolutional Neural Network (CNN)*," 2022 3rd International Conference on Computer Science and Management Technology (ICCSMT), Shanghai, China, 2022, pp. 292-295.

[04] D. R. Patel and R. K. Agrawal, "*Face Mask Detection and Recognition using OpenCV and Keras*," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Nagpur, India, 2020, pp. 1-5.

[05] S. Qi, X. Zuo, W. Feng and I. G. Naveen, "*Face Recognition Model Based On MTCNN And Facenet*," 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkur, Karnataka, India, 2022, pp. 1-5.

[06] P. Jain and R. Roy, "*Face Recognition with Mask Detection using Deep Learning and OpenCV",* 2021 5th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2021, pp. 442-446.

[07] Khan, Soumya & Sengupta, Diganta & Ghosh, Anupam & Chaudhuri, Atal, "*MTCNN++: A CNN-based face detection algorithm inspired by MTCNN*" , 2023, pp. 1-19.

[08] Wang Xiang, "*The Research and Analysis of Different Face Recognition Algorithms*", The International Conference on Computing Innovation and Applied Physics (CONF-CIAP), 2022, vol. 2386.

[09] Sun Y, Ren Z, Zheng W., "*Research on Face Recognition Algorithms on Image Processing*" , Comput Intell Neurosci, 2022.

[10] L. Li, X. Mu, S. Li and H. Peng, "*A Review of Face Recognition Technology*", in IEEE Access, vol. 8, pp. 139110-139120, 2020.

[11] G. Singh and A. K. Goel, "*Face Detection and Recognition System using Digital Image Processing*," 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 2020, pp. 348-352.

[12] P. J. Thilaga, B. A. Khan, A. A. Jones and N. K. Kumar, "*Modern Face Recognition with Deep Learning*", Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2021, pp. 1947-1951.

[13] M. Jha, A. Tiwari, M. Himansh and V. M. Manikandan, "*Face Recognition: Recent Advancements and Research Challenges*," 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022, pp. 1-6

[14]R. Sharma, V. K. Sharma and A. Singh, "A Review Paper on Facial Recognition Techniques," *2023 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India, 2021, pp. 617-621.