Vol. 7, No. 2 | July – December 2024

Robust Face Detection and Identification under Occlusion using MTCNN and RESNET50

Eiman Wahab[1*], Wajeeha Shafique[1], Habiba Amir[1], Sameena Javaid[1*] and Muhammad Marouf[1]

Abstract:

In today's rapidly evolving world, where technology is progressing swiftly, there is an increasing demand for facial recognition systems. Technologies are similar to digital forensics in that they can recognize people by scanning faces. However, one key problem they confront is dealing with covered or occluded faces, which might restrict recognition of faces in real-world situations. To overcome this issue, we created a system that is capable of identifying individuals even when their faces are veiled. We used the face detector algorithm called Multi-Task Cascaded Convolutional Neural Network (MTCNN) for face detection with 99.8% accuracy. Further we have conducted feature extraction and pre-processing on our self-created dataset. Dataset consists of 1,513 images, including 612 images without occlusion, 176 with upper occlusions (glasses, sunglasses), and 725 with lower occlusions (masks, niqabs). After data augmentation, the dataset expanded to 7,565 images, with 3,060 non-occluded, 880 upper occluded, and 3,625 lower occluded images. Our project utilizes the power of deep learning model: Residual Network (ResNet50), the form of deep neural network architectures well-suited for the job of features extraction. These features are matched by using Cosine similarity with accuracy of 92%. By leveraging the capabilities in the deep learning algorithms, this project provides a robust solution for automating the recognition of partially occluded faces.

Keywords: *Face recognition; Partially occluded faces; Face detection; Face Biometric; ResNet50; MTCNN;*

1. Introduction

Due to the demand for computer technology, daily tasks are increasingly being handled electronically rather than with pen and paper or face-to-face. With the development of computer technology, and growing need for user recognition and authentication that is swift and precise [1]. Nowadays, software's that has biometric recognition are becoming more significant in security, management, and commercial. Biometric systems include fingerprints, iris identification, retinal examining, voice recognition, and facial landmarks recognition [2][3]. Because it focuses on detection, identification, and verification [4].

Convolutional Neural Network (CNN) has been used for facial recognition because of their high frequency and positive recognition rate[1]. It is being utilized by law enforcement organizations, airport security, and public spaces equipped with surveillance cameras [4].

Modern Deep Learning (DL) systems have been shown to be practically faultless face detectors, outperforming human abilities in

[1] Department of Computer Sciences, Bahria University Karachi Campus, 13-National Stadium Road, Karachi, Pakistan

[*] *Corresponding Author*: eiman.wahab00@gmail.com; sameenajaved.bukc@bahria.edu.pk

this domain. However, it remains a difficult problem, especially when it happens to recognize faces under various forms of occlusion. Occlusion can occur due to a variety of factors, including facial accessories such as masks, scarves around the face, eyeglasses, hats, or caps, as well as changes in lighting conditions [5][6]. Face detection and recognition are essential elements of facial recognition systems, and they both play critical roles in the procedure.

1.1. Face Detection

Automatic identification of faces is crucial, it is the computer vision technique for locating or detecting faces within images or video streams [5]. The edge of the face is typically marked with a bounding box or other visual indicator. A rectangular region that surrounds the face is referred to as the bounding box.

Face detection frameworks are classified into two categories: those that rely on CNN and are known as Two stage detectors that undergoes from CNN layers, such as Multi-Task Cascaded Convolutional Neural Network (MTCNN), Region-based Convolutional Neural Network (RCNN), RetinaFace, and fast RCNN. The second classification is one-stage detectors that do not rely on CNN, that includes the Single Shot Multi-Box Detector (SSD), You Only Look Once (YOLO), Open-Source Computer Vision Library (Open CV), and Haar cascade Face Detector. The performance of two stage detectors is better than the one stage detectors but demand more processing time; on the other side, one-stage detectors take lesser processing time but produce less exact detection findings [6].

Face detection is a crucial aspect of computer vision and pattern recognition, gaining significant attention in recent years. It serves as the foundational step for various facial-based applications, including face verification, recognition, clustering, security systems, photography, and video analysis. [7].

Following is some of the major challenges in detection of face:

- **Pose variation**: Faces can be in different poses in the images or video streams, such as looking up, down, right, or left. In different poses, it is very challenging for detection algorithms to detect faces.
- **Illumination variation**: Faces can be illuminated in several ways, such as by artificial light, natural light, or shadows. This illumination variance is also very challenging for detection algorithms to detect faces.
- **Occlusion**: Faces can be partially or fully occluded by things such as masks, hair, pair of glasses, or hats that create hurdles for detection algorithms to detect faces.

1.2. Face Recognition

Recognition of faces is a technology that allows computers to recognize and recall the faces of individuals. Consider it a digital way of remembering someone you've already met. The primary aim of this type of face recognition is to figure out the identity of the number of people depicted in each scenario by using their unique facial characteristics that distinguish them from others. Face recognition uses a stored database of faces to identify or verify one or more people [8]. To identify a person, it entails capturing and examining facial patterns and traits from digital photos or video frames.

The process often begins with face detection, in which the system locates and recognizes the occurrence of faces within a frame provided. And this basic phase helps to separate the face or faces from the rest of the image. After detecting the face, the system extracts facial features such as the distance of eyes, and distance of nose, distance between both of them and these are the characteristics which are subsequently compared to a database of already detected and identified faces to see if they match.

However, when confronted with problems such as greater pose variability, fluctuating illumination variation, low lights or night resolution, diverse expressions on the face and obstruction, recognition systems for faces continue to fall short [2].

The logic flow for face recognition using neural networks is depicted in Figure 1. There are numerous face detection techniques available because this is an evolving area of research in computer vision. The software pre-processes each and every individual face in the image. When a face is spotted, it generates a normalized input for CNN. and then the neural network serves to extract feature to produce faces of every person which is dimensionally low in representation [9].

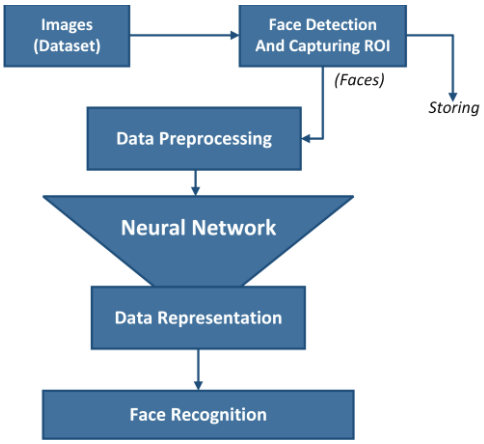

Fig. 1. *Flow of face recognition with neural network.*

1.3. Face Occlusion

Face occlusion is the partial concealment or obstruction of the face, that makes this task nearly impossible for human observers or facial recognition software to precisely identify or recognize the person. Because we have no previous information of the covered area, that might be present at any part and at any inch or form in a face, facial occlusion is regarded as one of the most intractable difficulties [4].

Some challenges are faced when the facial feature of a person is partially veiled including high processing cost, resistance towards image modifications and occlusions, and learning perceptive representations of occluded faces, are currently being thoroughly investigated and analyzed. To remove the occlusion of obstructed faces and to identify individuals whether they are students marking attendance by facial recognition system, shopkeepers from security cameras, or criminals attacking citizens, the occluded face recognition system is very much needed [3].

As a result, recognizing faces under occlusion remains a difficult problem. Facial occlusion occurs facial characteristics can be partially or entirely obscured by ordinary things such as masks, glasses, scarves, and numerous accessories. Masks have become a key difficulty because of public health laws, covering critical facial parts such as the mouth and nose. Furthermore, eyewear, such as glasses or sunglasses, can considerably obstruct detection by obscuring the eyes, which are crucial in identification. Scarves, hats, and other garment items may make the task more difficult by obstructing areas of the face, particularly around the eyes [10].

Types of occlusions for testing Occluded Face Recognition (OFR) which are mentioned in Figure 2.

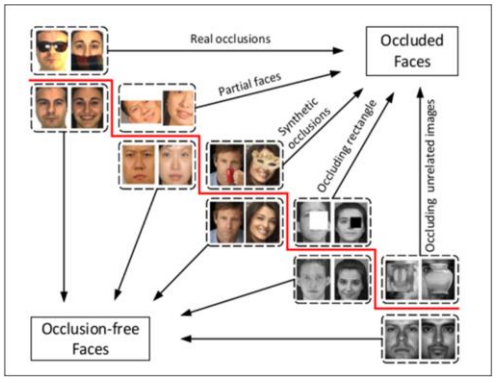

Fig. 2. *Different types of occlusions* [2].

2. Literature Review

2.1. Background

The one task that is considered the most critical task in computer vision is recognition of faces. Human faces offer much better features to identify a person than other commonly used biometric technologies like iris or fingerprints [11]. However, FR algorithms are affected by face disorders like occlusion and changes in lighting and facial contours. Occluded face recognition is the task

of identifying individuals whose faces are partially or fully obscured by objects such as masks, niqab, makeup, or sunglasses. This is a challenging problem for traditional facial recognition methods, which typically rely on the full face to extract features for identification.

In computer vision, detection of faces has been extensively and deeply analysed and studied. However, before the year of 2000, although numerous published papers, but the applications of the detection of faces needed efforts to make a perfect one. There was no existing system until the revolutionary work of Viola and Jones [7]. Viola-Jones is a simple, fast, and powerful object detection algorithm that has proven as the most effective in object detection applications, that includes face detection, walkie-talkie detection, and automobile detection. Since it is a simple algorithm, it is one of the most popular algorithms for object detection in real-time. However, it is sensitive to illumination and viewpoint changes, and is fooled by objects with features similar to that of the target object.

Deep learning techniques has now made the face detection possible, which often outperform conventional computer vision techniques. "Face Detection with End-to-End Integration of a ConvNet and a 3Dimensional Model" introduced a strategy for detection of faces that integrates a convolutional network with a model of face in a complete multi-task framework of learning [12]. In a recent paper [13] [14] used the FAST R-CNN, one of the most advanced generic object detectors, and the results were promising. There has also been a lot of efforts done to improve and make the architecture faster.

CNN has been implemented on RPN and also on the Faster R-CNN to achieve complete optimization [15]. The combination of Faster R-CNN face recognition model and ResNets by [16] Wan et al. has achieved major improvements in face recognition benchmarks.

2.2. Related Work

During the pandemic, mask usage became a major challenge for face recognition technologies. The authors proposed a mask-based face recognition technique in [17] . The authors presented three new datasets, including RMFRD, which achieved 95% accuracy, surpassing industry benchmarks. This work highlighted the need to adapt face recognition to changing societal needs.

In [18], they took a step-by-step approach to face detection and classification. They used features selection and deep learning to figure out which faces were masked and divided them into three groups: masks, covered faces, and unmasked faces.

In [9], the challenges, and techniques of face recognition for mobile devices were examined. Various face recognition techniques were compared, such as DNN, Eigenfaces (Eigen), Fisherfaces, and Local Binary Pattern such as FaceNet and OpenFace. This was conducted based on the dataset of LFW (Located Faces in the Wild) which proved the applicability of DL for mobile recognition of faces.

In recent years, the multi-face recognition is the field that has become increasingly prominent. The authors of [11] provided an in-depth analysis of the most current multi-face recognition approaches. They highlighted the general development pipeline and comparative datasets commonly employed in MFR studies. They provided insight into the difficulties in this field directions that has a great potential for future research.

Several reports have been published about face recognition with occlusion. The proposed techniques, such as dynamic feature subset selection (DSP) or occlusion recovery (OR), were intended to enhance face recognition in the context of occlusion. And it did emphasize the significance of resilient algorithms for dealing with occluded faces.

In [2], RetinaFace was introduced as a one-stage face detector. Five key contributions were identified in the paper, such as facial landmark annotation and three-dimensional shape prediction. The ability of RetinaFace to surpass existing methods in terms of face localization on the dataset WIDER FACE marked a significant development in face detection.

In [19], experiments with open-source face related datasets (Label Face in the Wild and YouTube Face database) were presented. and we can say that emphasis was placed on a model with concurrent SoftMax supervision and center loss of data. It demonstrated that the proposed approach was superior to existing methods for face verification and identification.

3. Methodology

Framework consists of several stages. Each stage has its own methodology. The camera captures the images that are detected by algorithms used. It then undergoes features extraction process from model of deep learning. CNNs are sophisticated models capable of capturing valuable information. Using a CNN that is already trained on a large data as a facial feature extractor is likely to be effective as a building block for training CNN to remove occlusion and detect individual faces. The proposed CNN architecture relies upon a highly deep and powerful face recognition CNN that can extract facial features in a distinctive and robust manner. Some CNN models have been effectively trained for recognition tasks of human faces [20]. We used the power and capabilities of Residual Network (ResNet50) in our project.

3.1. Dataset Collection

In the initial phase we create our own dataset according to the needs of the research. We first developed a consent form ensuring ethical considerations of our data collection and actively sought individuals in our surroundings including our families, friends and university who are willing to participate in our dataset collection, and after obtaining their consent through signed agreements, we proceeded to capture a diverse range of facial images. The diversity includes unveiled, veiled, and different types of facial accessories such as glasses, sunglasses, niqabs and head scarves.

The next step of dataset creation was to organize the extensive collection, we created dedicated folders for each participant with their respective names. Each folder contains a variety of images of the person, labelled chronologically. We ensured that each image is annotated, and well-organized folders are formed which will be used for extraction of features and matching of features.

In order to ensure that our model is exposed to a variety of situations, including partial face coverage from accessories (such as masks, glasses, and scarves), differences in lighting, and a wide range of ethnicities, age groups, and genders, our dataset comprises a wide range of facial photos with varied levels of occlusion. Because faces are rarely caught in perfect conditions in real-world applications, this diversity is essential to ensuring that the model generalizes properly.

TABLE I. Total Count of Images in Dataset

Total Count of Images	**Type of Occlusion**			
	No	***Upper (glasses, sunglasses)***	***Lower (mask, niqab)***	***Total Count***
	612	176	725	1513

TABLE II. Count of Images in Dataset after Data Augmentation

Total Count of Images	**Type of Occlusion**			
	No	***Upper (glasses, sunglasses)***	***Lower (mask, niqab)***	***Total Count***
	3060	880	3625	7565

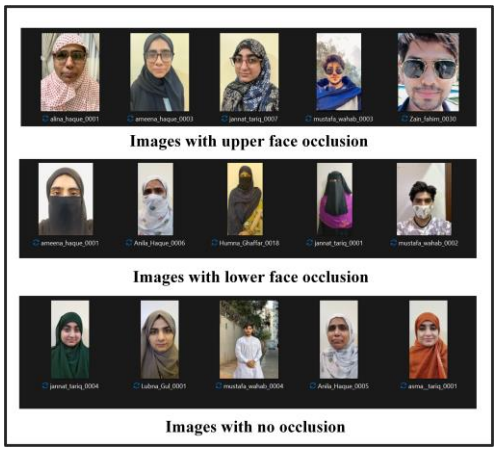

Fig. 3. *Images from dataset*

To address potential dataset limitations on size and improvement in reliability of the architecture, we increased the size of dataset by 100 images in every category. Despite this increase, the results and findings are consistent and demonstrating performance stable. This method ensures that the system can handle a variety of occlusions while retaining recognition accuracy.

3.2. Face Detection Using MTCNN

Face identification algorithms often start by recognizing the borders of a face. Once the edges have been recognized, the algorithm can determine the pose and dimension of the face. A bounding box or other visual signal is commonly used to denote the position. A rectangular zone that surrounds the face is known as the bounding box. For that purpose, well- suited and highly responsible framework is used MTCNN.

MTCNN is a deep learning CNN detection algorithm. It helps to identify faces and facial landmarks; it minimizes variations introduced by head pose or image conditions. It is among the most widely used face detectors that are used today that provide robust accuracy [21].

The first stage is called Proposal Network (P-Net) in which the image is scaled several times to detect faces of various sizes. The first neural network searches the images and performs the first detection. It has a low detection threshold, which causes it to identify numerous false positives even after NMS (Non-Maximum Suppression), however this is designed. The proposed regions (which contain many false positives) are taken by the second network called Refinement Network (R-Net); it filters detections (together with NMS) to provide very precise bounding boxes. The third layer, called Output Network (O-Net), refines the detected faces in bounding boxes.

This method not only detects faces but also produces very accurate bounding boxes. MTCNN also helps in defining the Region of Interest (ROI) of landmarks on faces features, within face region that have been detected. MTCNN systematically analyse each image, identifies facial landmarks, and localizes faces in the frame.

It is the most important step in the face recognition systems. This step is needed for features extraction to help the model identify the region of maximum features present. When the data of 150 images was detected using MTCNN, the percentage of detected faces was 99.8%. Results of MTCNN are shown in figure 4.

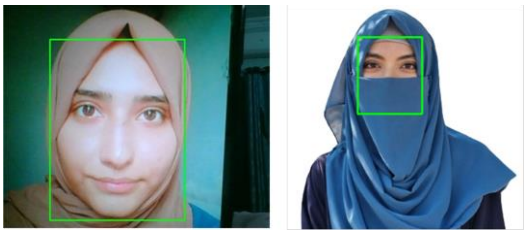

Fig. 4. *Bounding box by MTCNN*

Following figure 5 is the images detected, cropped, and stored by MTCNN:

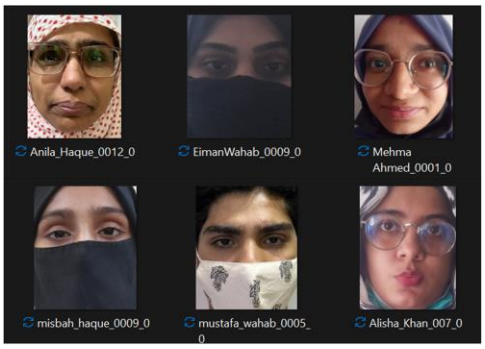

Fig. 5. *Detected faces with MTCNN*

3.3. Feature Extraction Using ResNet50

We used ResNet50 for extracting features. The residual block in Residual Networks can be used as CNN layer that is by learning to the residual stream of zero or ResNets block with single layer that is by learning to zero the transient stream. The general residual architecture offers the flexibility of learning anything in between by constantly repeating the residual block. The features are extracted from ResNet50 and used in the features matching stage. Each 224x224 image is thus represented by a 4,096-dimensional vector to represent every feature [10].

This is an important task to remove occlusion and focus on the available areas of the face. That is why we aimed to use ResNet50 due to the outstanding performance even on the change in illumination, facial expression, and facial occlusion. ResNet50 has 50 layers designed to reduce the extent of parameters and features matrix multiplications, resulting in a "bottleneck". In recent times, ResNets have overtaken other models in complex computer vision tasks. The ResNets network is composed of 34 plain layer architecture on the basis of VGG-19. ResNets consist of Residual Blocks. A residual block learns the residual (difference) between the desired output and the input rather than attempting to learn the desired output directly from the input. The ResNets design introduces the simple concept of combining the output of a sequence of convolution blocks with an intermediate input [22].

We employ ResNet50 for feature extraction because it can handle complicated visual patterns, particularly in occluded face recognition. The model, which has 50 layers and residual connections, addresses the vanishing gradient problem. We trained with a learning rate of 0.0001, a batch size of 32, and the Adam optimizer. To avoid overfitting, regularization approaches such as a 0.5 dropout rate were used. To maximize computational efficiency, the model was trained for 50 epochs and then stopped early.

3.4. Face Recognition By Pattern Matching

When the model is trained, it can recognize faces by extracting and matching stored features from stored images. During training, the model learned to capture discriminative facial features that are crucial for distinguishing one person from another, even when occlusions or changes in lighting and location exist. These learned characteristics serve as separate representations of each person's face.

The approach used is cosine similarity that collects comparable features from the incoming image and then compares them to the already saved representations in the datastore to recognize the face of individual [10]. It is a Machine Learning (ML) metric that takes features of two images as input and returns the distance between these features to determine the probability that they are the same person or not. The distance between two facial features is calculated using the formula of cosine distance. that evaluates the similarity of two vectors in an inner product. Facial features produced by deep learning models are represented as vectors in a multidimensional space. Cosine similarity can then be used to calculate the angle between these vectors, yielding a measure of similarity.

$$cos\,\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

Cosine similarity works well when the amplitude of the feature vectors is less essential than their direction. It is often employed in face recognition algorithms that prioritize the relative orientation of facial characteristics above their absolute values. Using cosine similarity, when we matched results from our stored data to tested data, the accuracy was 92% with the execution time of 1.3 seconds. The result in similarity percentage along with person's name is shown above each bounding box.

Fig. 6. *Predicted output with similarity score.*

3.5. Framework

In a directory the images of the collected dataset are stored. The facial features from these images were detected by the renowned algorithm for face detection: Multi-Task Cascaded Neural Network (MTCNN). And most importantly ROI (Region of Interest) is plotted for further features extraction.

In the next step, these detected faces are taken input by deep learning model for features extraction. The model we have used for this purpose is the ResNet50 model. The extracted features are stored in database for features matching.

Our work is novel in that it focuses on face recognition in occluded circumstances, which is a challenging and underexplored field. Unlike existing methods, which often perform well in ideal situations, our solution addresses the particular challenges given by occlusion, which has a major impact on the accuracy of standard face recognition models. By using MTCNN for face detection and ResNet50 for feature extraction.

At real-time, the input of camera capture that has facial features present in it is detected using same MTCNN model. After the detection the facial features are extracted by Resnet50. These extracted features are then matched with already saved features by Cosine Similarity. The System's framework is illustrated in figure 7.

4. Results

We conducted an experiment using volunteer participants to assess the system's performance under various occlusion conditions. Each volunteer's facial features were collected in three different scenarios: no occlusion, upper occlusion, and lower occlusion. The acquired data was evaluated to determine the detection of individuals under these various situations.

TABLE III. MTCNN Results of Detection

Total 100 images		
Type of Occlusion	***Successful Detection***	***Failed Detection***
Without Occlusion	1	0
Upper Occlusion	1	0
Lower Occlusion	1	0

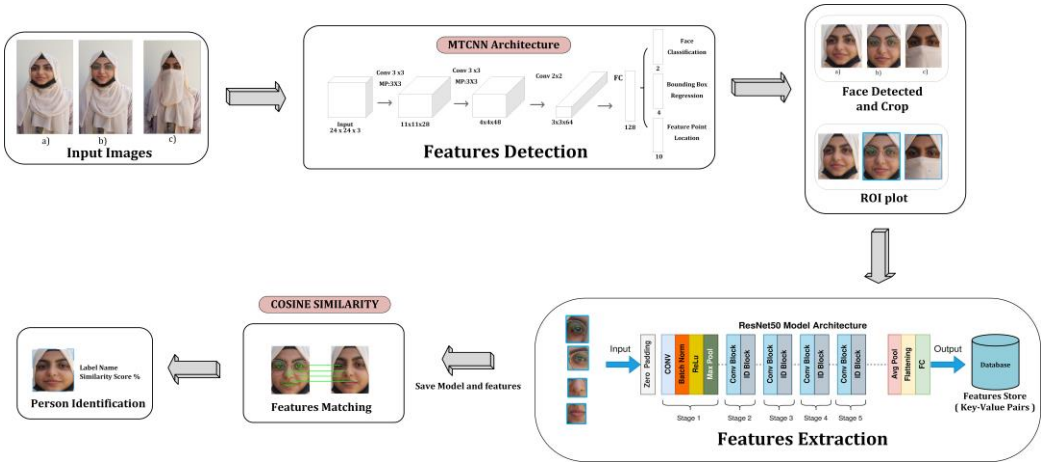

Fig. 7. *Framework for facial partial occulsion detection*

The acquired data was then evaluated to determine the system's accuracy in recognizing individuals under these various situations.

TABLE IV. Accuracy Results on different type of Occlusions

Participant	Without Occlusion	Upper Occlusion	Lower Occlusion
01	0.97	0.98	0.97
02	1	0.97	0.99
03	1	1	0.98
04	1	1	0.97
05	0.98	0.98	1
06	1	0.96	1
07	0.98	1	0.96
08	1	0.98	0.95
09	1	0.98	0.98
10	0.99	0.99	0.98
11	1	0.99	1
12	1	0.98	0.98
13	0.99	0.97	0.97
14	1	1	0.99
15	1	1	1
16	0.99	0.98	0.97
17	0.98	0.96	0.98
18	1	1	0.9
19	0.99	0.97	0.97
20	1	1	1
21	0.99	1	0.99
22	1	0.99	1
23	0.96	0.87	0.89
24	1	0.97	0.87
25	0.97	0.98	0.98
26	1	1	0.9
27	0.89	0.87	0.89
28	0.98	0.98	1
29	1	0.96	1
30	0.98	1	0.96
31	1	0.98	0.95
32	1	0.98	0.98
33	0.99	0.99	0.98
34	0.96	0.98	0.98
35	1	0.98	0.99

To visually represent the accuracy average of the experiment, the collected data is now plotted on a line graph shown in figure 8. The

graph plotted the average accuracy on y-axis and the types of occlusions are shown on x-axis.

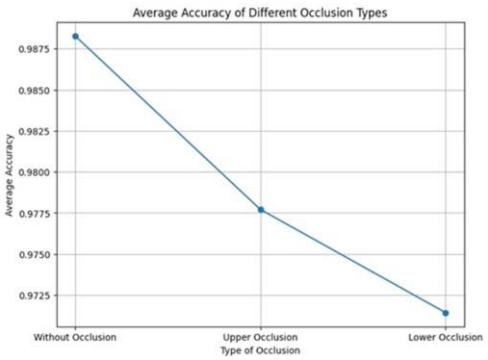

Fig. 8. *Average Accuracy on different types of occlusions.*

Upon analyzing the weak performance areas, it was observed that the model showed slight dips in accuracy in the presence of lower occlusions (such as masks and niqabs). This is likely due to the occlusion of key facial features like the mouth and chin, which are essential for accurate face recognition. This challenge is consistent with findings in other studies [2][4] and [18], which reported that occlusions affecting the lower half of the face hinder recognition performance. These weak performance areas suggest potential improvements, such as integrating additional facial cues or utilizing advanced algorithms for occlusion handling, which could help boost performance in such scenarios.

5. Comparative study of models

Comparative research was held to perform analysis on different models. It began with the input of image or real time camera capture that has facial features present in it. The foremost key step in the process of facial identification is Face Detection. We were not only capturing the bare faces, but the aim was to capture the available facial features in the presence of occlusion. That is the reason we performed evaluation between the two face detection algorithms namely Haar Cascade and MTCNN.

The Haar Cascade for face identification uses a set of trained classifiers, each of which is trained to distinguish between parts of an image containing faces and those that do not. Adaptive Boosting (Adaboost) which is also a machine learning based model that selects a limited number of the most relevant Haar features and assigns them weights. When the data of 150 images was detected using Haar Cascade, the percentage of detected faces was 91.89%.While MTCNN showed 100% accuracy. MTCNN was chosen for detection because of its better results.

TABLE V. Comparison of Haar Cascade and MTCNN Performance Results 150 Image

Total 150 images		
Model Name	***Haar Cascade***	***MTCNN***
Face Detected	137	150
Average Processing Time (seconds)	1.659	1.543
Overall Accuracy	0.918	1.0
Overall Data Loss	0.4	0.0

In the next step, these detected faces are taken input by our deep learning models for extraction of features. It is an important stage in face identification system that tries to extract several distinguishable characteristics that can be used for representation and the learning essential facial features of eyes, nose, mouth, and texture [19]. These models we used for evaluation purposes were VGG-16, VGG-19 and ResNet50 model. The extracted features were stored for features matching. ResNet50 was chosen for features extraction because of its better results in processing time and with the features matching model.

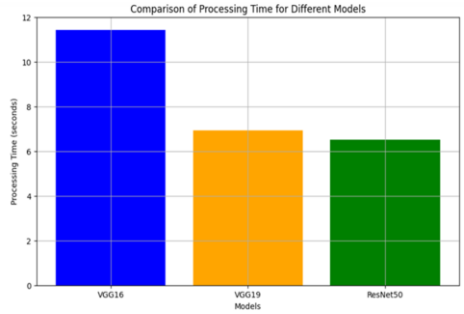

Fig. 9. *Difference in processing time of each model*

Next the new input is taken by camera capture that undergoes with similar face detection technique. After the detection the facial features are extracted by the models used. These extracted features are then matched with already saved features. Facial feature matching usually entails comparing specific facial characteristics or landmarks between two faces to determine their similarity. This method is crucial to face recognition systems, which aim to identify or authenticate individuals using facial information. The new test image is matched with the saved features using three different approaches.

This features matching process was evaluated using Cosine Similarity, Manhattan Distance and Euclidean Distance. Using Euclidean distance, when we matched results from our stored data to tested data, the accuracy was 89% with the execution time of 1.1 seconds. Using Manhattan distance, when we matched results from our stored data to tested data, the accuracy was 90% with the execution time of 1.49 seconds. Cosine Similarity is chosen for features matching because of its better results of 92% accuracy. The comparison of the three models can be seen in the table below.

TABLE VI. Comparison of Processing Time and Accuracy of Cosine Similarity, Euclidean Distance and Manhattan Distance

Similarity Metrics	**Processing time (seconds)**	**Accuracy**
Cosine Similarity	1.49	0.92
Euclidean Distance	1.1	0.89
Manhattan Distance	1.49	0.90

The system stores a record of extracted facial features in the database. When the user sends a real-time facial image via webcam, the face detection algorithm is executed, followed by matching the extracted features from the image with those stored in the database. Different use cases are considered, including scenarios with upper occlusion caused by glasses and sunglasses whereas lower face occlusions caused by masks and niqab, as well as situations involving multiple faces. The following are the use cases of our system.

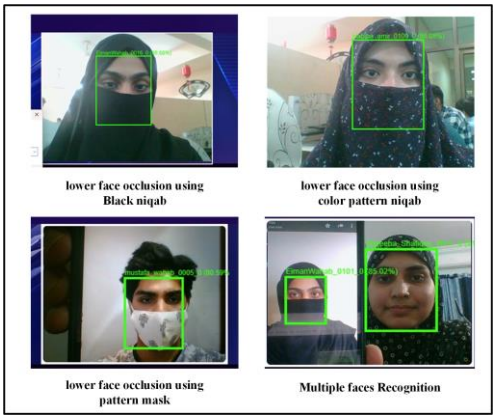

Fig. 10. *Face Recognition with lower occlusion*

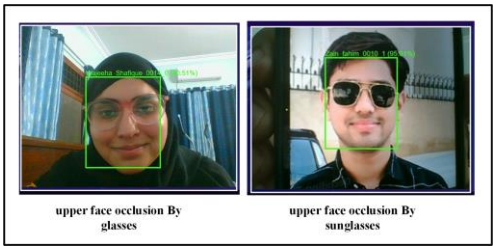

Fig. 11. *Face Recognition with upper occlusion*

Strategies for reducing processing time in real-time use cases include employing efficient model designs such as MobileNets or

EfficientNet, using hardware acceleration (GPUs, TPUs, FPGAs), and implementing techniques such as model quantization, pruning, and distillation. Edge computing can reduce latency by processing data locally, and optimizing data pipelines, caching, and lowering input data resolution all help to speed up processing. These combined approaches improve system efficiency while preserving performance.

6. Conclusion

We have successfully overcome the challenges of traditional face recognition system when the facial features are partially covered with any accessories like glasses, veil, or mask. By utilizing the MTCNN as face detector we developed a robust system that is able to correctly identify faces in the frame at real-time. Furthermore, ResNet50 has played an important role in feature extraction, allowing the system to collect discriminating facial features required for identification. In addition, the adoption of the Flask framework has made it easier to create a user-friendly web application with seamless interaction and accessibility across multiple devices and browsers. Users may quickly access camera functionality, start face recognition, and receive real-time feedback on identified faces and recognized individuals.

6.1. Future Directions:

As we have implemented our system and achieved the goal of handling the occlusion on lower and upper part of the face. We have aimed to add more types and degrees of occlusions to make our system more efficient. We will use different obstacles to obscure the different parts of face by adding more variety in our dataset. This would help to make the face recognition system more reliable. We will make our system applicable to different environmental conditions. We also aim to use GANs (Generative Adversarial Networks) to enhance the variation of data. We are hopeful that this project will lead us to greater opportunities. Additionally, we will explore the integration of advanced optimization techniques, such as using a lightweight model and model quantization, to improve processing speed for real-time applications.

AUTHOR CONTRIBUTION

Eiman Wahab: Methodology, Software, Conceptualization, Visualization and Data Curation. Wajeeha Shafique: Software, Validation, Formal analysis and Data Curation. Habiba Amir: Resources, Data Curation, Writing- Reviewing and Editing. Sameena Javaid: Supervision, formal analysis. Muhammad Marouf: Co-Supervision and report review.

DATA AVAILABILTY STATEMENT

The datasets generated and analyzed during the current study are not publicly available due to privacy concerns but can be available on reasonable request.

CONFLICT OF INTEREST

No Potential conflict of interest was reported by authors for this research.

FUNDING

No funding was received.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to our research supervisor, Ma'am Sameena Javaid, for her invaluable advice and guidance throughout the development of the research. We would also like to express our gratitude to all the participants who volunteered in dataset collection.

In addition, we are deeply thankful to our loving parents, whose help and encouragement have led us to success.

REFERENCES

[1] A. M. P, S. N, and M. K, "Face Recognition using CNN: A Systematic Review," International Journal of Engineering Research & Technology, vol. 11, no. 6, Jun. 2022, doi: 10.17577/IJERTV11IS060075.

[2] D. Zeng, R. Veldhuis, and L. Spreeuwers, "A survey of face recognition techniques under occlusion," IET Biometrics, vol. 10, no. 6, pp. 581–606, 2021, doi: 10.1049/bme2.12029.

[3] C. Gopalan and Ms. S. Suchitra, "A Survey on Face Dtection," International journal of recent advances in engineering and technologies,

published by IRD, vol. 1, pp. 113–118, Jan. 2013.

[4] M. Oloyede, G. Hancke, and H. Myburgh, "A review on face recognition systems: recent approaches and challenges," Multimedia Tools and Applications, vol. 79, Oct. 2020, doi: 10.1007/s11042-020-09261-2.

[5] M. K. Meena and H. K. Meena, "A Literature Survey of Face Recognition Under Different Occlusion Conditions," in 2022 IEEE Region 10 Symposium (TENSYMP), Jul. 2022, pp. 1–6. doi: 10.1109/TENSYMP54529.2022.9864502.

[6] H. Moqbel and M. Parameswaran, "Occluded Facial Recognition for Surviellance Using Deep Learning," in 2022 8th International Conference on Virtual Reality (ICVR), Nanjing, China: IEEE, May 2022, pp. 459–466. doi: 10.1109/ICVR55215.2022.9848162.

[7] X. Sun, P. Wu, and S. C. H. Hoi, "Face detection using deep learning: An improved faster RCNN approach," Neurocomputing, vol. 299, pp. 42–50, Jul. 2018, doi: 10.1016/j.neucom.2018.03.030.

[8] "(PDF) Face Recognition from a Single Image Per Person: A Survey." Accessed: Sep. 10, 2023. [Online]. Available: https://www.researchgate.net/publication/222672070_Face_Recognition_from_a_Single_Image_Per_Person_A_Survey

[9] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "OpenFace: A general-purpose face recognition library with mobile applications".

[10] Q. Jin, C. Mu, L. Tian, and F. Ran, "A Region Generation based Model for Occluded Face Detection," Procedia Computer Science, vol. 174, pp. 454–462, Jan. 2020, doi: 10.1016/j.procs.2020.06.114.

[11] A. Alzu'bi, F. Albalas, T. AL-Hadhrami, L. B. Younis, and A. Bashayreh, "Masked Face Recognition Using Deep Learning: A Review," Electronics, vol. 10, no. 21, Art. no. 21, Jan. 2021, doi: 10.3390/electronics10212666.

[12] Y. Li, B. Sun, T. Wu, and Y. Wang, "Face Detection with End-to-End Integration of a ConvNet and a 3D Model," Aug. 29, 2016, arXiv: arXiv:1606.00850. doi: 10.48550/arXiv.1606.00850.

[13] H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," Jun. 10, 2016, arXiv: arXiv:1606.03473. doi: 10.48550/arXiv.1606.03473.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jan. 06, 2016, arXiv: arXiv:1506.01497. doi: 10.48550/arXiv.1506.01497.

[15] H. Qin, J. Yan, X. Li, and X. Hu, "Joint Training of Cascaded CNN for Face Detection".

[16] S. Wan, Z. Chen, T. Zhang, B. Zhang, and K. Wong, "Bootstrapping Face Detection with Hard Negative Examples," Aug. 07, 2016, arXiv: arXiv:1608.02236. doi: 10.48550/arXiv.1608.02236.

[17] "Wang: Masked face recognition dataset and application - Google Scholar." Accessed: Sep. 10, 2023. [Online]. doi https://doi.org/10.48550/arXiv.2003.09093

[18] H. Al-Dmour, A. Tareef, A. M. Alkalbani, A. Hammouri, and B. Alrahmani, "Masked Face Detection and Recognition System Based on Deep Learning Algorithms," JAIT, vol. 14, no. 2, pp. 224–232, 2023, doi: 10.12720/jait.14.2.224-232.

[19] A. Bansal, A. Nanduri, C. Castillo, R. Ranjan, and R. Chellappa, "UMDFaces: An Annotated Face Dataset for Training Deep Networks," May 21, 2017, arXiv: arXiv:1611.01484. doi: 10.48550/arXiv.1611.01484.

[20] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "RetinaFace: Single-stage Dense Face Localisation in the Wild," May 04, 2019, arXiv: arXiv:1905.00641. doi: 10.48550/arXiv.1905.00641.

[21] B. Houshmand and N. Khan, "Facial Expression Recognition Under Partial Occlusion from Virtual Reality Headsets based on Transfer Learning," Aug. 12, 2020, arXiv: arXiv:2008.05563. doi: 10.48550/arXiv.2008.05563.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.