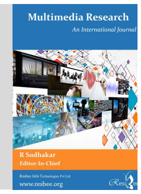
Vol.7 No.4 Oct 2024

Multimedia Research

Received 5 August, Revised 1 November, Accepted 6 November



Development of a Real-Time Auto Encoder Facial Occlusion Recognition System Framework

Afolabi Awodeyi
Department of Computer Engineering Veritas University Abuja, Nigeria
awodeyia@veritas.edu.ng

Philip Asuquo
Department of Computer Engineering University of Uyo, Nigeria
asuquophilip@gmail.com

Constance Kalu
Department of Computer Engineering University of Uyo, Nigeria
constancekalu@uniuyo.edu.ng

Abstract: The inability to recognize occluded faces due to nose masks has always been a problem that presents a major challenge in facial recognition systems. Face identification under occlusion which involves a nose mask has less research coverage when compared to challenges such as variation in pose and expressions. This research was carried out to solve the problem of limited capacity to recognize faces under occlusion due to nose masks and poor lighting conditions. The system algorithm is divided into the training and recognition system. The training system has a training set of 500 images belonging to 100 people which were scrapped from the internet. The train system consisted of image acquisition; consisting of image augmentation which increased the model's capacity to generalize what it had learned, the feature extraction; which involved the extraction of annotated features such as the eye, nose, forehead, and chin. The recognition system is made up of image acquisition, preprocessing, feature extraction, classification, and recognition. The preprocessing made use of a canny edge detector which is a preprocessing tool used to detect edges in images in a very robust manner. The feature extraction phase is the next phase which makes use of the Auto Encoder CNN; implemented using face with and without face masks in real-time on a test set of 70 persons. The autoencoder had an average match rate of 61.67% for faces without nose masks and 56.86% for faces with nose masks. This research has proven that facial recognition systems can overcome the challenge of face occlusions caused by nose masks. This research has proven that facial recognition systems can overcome the challenge of face occlusions caused by nose mask occlusion which is a challenge from other research.

Keywords: Nose Mask, Occlusion, Augmentation, Annotated, Preprocessing, Canny Edge Detector, Auto Encoder, Surveillance.

Nomenclature

Abbreviation	Expansion
CNN	Convolutional Neural Network
MAFA	Masked Faces
RMFD	Real-World Masked Face Dataset
VR	Virtual Reality
DL	Deep Learning
MSE	Mean Squared Error
MTCNN	Multi-Task Cascaded Convolution Neural Network
SVM	Support Vector Machine
FCN	Fully Convolutional Network
NMS	Non-Max Suppression
FAR	False Acceptance Rate
FRR	False Rejection Rate

1. Introduction

A facial recognition system is a non-intrusive recognition system used for identifying the identities of individuals using faces. Facial recognition system is in a class of biometric identification software that is used to mathematically map facial features which is stored as a faceprint. This biometric identification software uses machine learning algorithms to compare live captures of images or digital images to the stored faceprint to verify identity. Face recognition systems employ computer algorithms to choose unique and distinguishing details on a person's face. These characteristics, such as eye distance or chin shape, are then converted into a mathematical representation and compared to data from other faces in a face recognition database. The information on a particular face, referred to as a face template, is limited to only those elements that can be utilized to distinguish one face from another [1].

https://doi.org/10.46253/j.mr.v7i4.a3

Facial recognition systems vary in their abilities when identifying faces under challenging conditions. These challenging conditions include poor lighting, low quality of the image resolution, and sub-optimal view angle (The image taken from above does not capture certain features of an unknown person).

In this paper, an Auto Encoder Model in the recognition system implements an algorithm for facial occlusion recognition. The Auto Encoder encodes annotated occluded face regions, compares with full faces stored in the database, and matches the occluded face with a face that has the minimum cosine distance for correct matching. Facial Occlusion Recognition systems can be helpful in areas where there is a high prevalence of occlusion [1, 4, 5, 7, 13].

The rest of this paper is organized as follows: section 2 discusses related work in facial recognition, Section 3, discusses the methodologies, section 4 presents the proposed approach, Performance Evaluation is presented in Section 5 and Section 6 concludes the paper.

2. Related Work

Several techniques [2,3,4,5] have been developed for facial recognition and detection systems with examples such as Viola-Jones, K-Nearest Neighbor, etc. New researchers have also used Convolution Neural Networks which is a more efficient technique because it maps out important features for facial recognition and occlusion recognition systems. Some existing research works are reviewed in this section below:

CNNs in [2] were used to recognize facial expressions in a context with severe occlusions. CNN's objective was to recognize people's facial expressions while wearing VR headsets that covered the upper half of their faces. The upper half of the face was purposefully occluded in the training dataset to accurately train neural networks for this environment, in which faces are significantly occluded. The neural network was able to focus on the lower half of the face as a result of this. A better accuracy rate was obtained which according to the researchers had a better accuracy rate than models trained with their entire faces. The results were based on the FER+ and Affect Net benchmark data sets. The results of their CNN models on the lower-half faces were up to 13% higher than other CNN models trained on the complete face, according to this study. The lower region of the face contains clues that can be used to properly predict facial expressions, according to this study.

Ryumina, E., *et al* (2021) [3]implemented a unique hybrid method for automatically detecting and recognizing the presence or absence of a protective mask on a person's face. It combined picture histograms with CNN visual features to convey information about pixel intensity. The Medical Mask Dataset was used to test their method, while the MAFA and RMFD databases were used for analysis. When compared to traditional CNNs on the MAFA and RMFD databases, the researchers' proposed hybrid technique raised the Unweighted Average Recalls of the recognition of the presence/absence of a protective mask on the human face by 0.96 percent and 1.32 percent, respectively. This proposed method could be expanded and applied to a variety of biometric applications, as well as computer vision, machine learning, and automatic face recognition [3].

Khan, S., *et al* (2019) [4]executed a framework for smart glasses that could recognize faces. The Machine Learning technique used was of high accuracy as compared to other techniques based on the researcher's analysis. Haar-like features were used for face detection. The detection rate of this method was 98% using 3099 features. Transfer learning of a trained CNN model was done for face recognition using AlexNet. An accuracy of 98.5% using 2500 variant images in a class was obtained [4].

Alagarsamy, S., et al (2020) [5] developed a smart recognition system for identifying criminals which were based on the features extracted. Convolution Neural Networks were utilized to assess the recognized images. Face recognition and verification were trained using images from a custom data set. By training the system with data, the TensorFlow framework was utilized to discover the faces of criminals. The criminal images were used to train the system for detection. This technique avoids the drawbacks of CNN end-to-end learning and the time consumption of very large-scale training data. Data sets containing YouTube faces were used for the evaluation of the proposed methodology. A face verification task was performed between two faces by calculating the similarities and differences between two objects. Features of the image were extracted based on the color, and intensity of the pixel, and finally, classification was performed. The suggested techniques delivered a better validation rate compared with the conventional techniques and required less time for processing. The burden of training the networks was also reduced [5].

Jaswal *et al.* (2014) [15] used the concept of CNN as a DL method for image classification. Aerial photograph remote sensing data (UC Merced Land Use Dataset) and scene photographs from the SUN database were used to test the algorithm. The algorithm's performance was evaluated using the MSE quality metric and classification accuracy. Based on MSE against the number of training epochs, a graphical representation of the experimental data was created. Quality metrics were used to analyze the experimental results, and the graphical depiction showed that the method CNN provided an average classification accuracy for all of the datasets evaluated. One of the many variants of the MSE for face and non-facial photos of 200 epochs is shown below.

The system could not identify individual faces with occlusion, affected by lighting conditions [15].

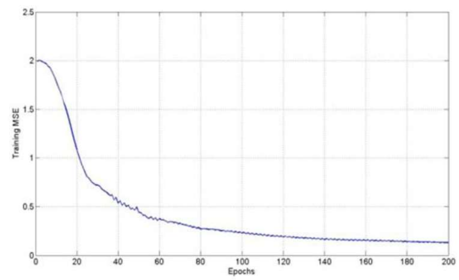


Fig. 1. *Variation of MSE for face vs. non-face images (200 epochs)*

Ejaz, M. S., and Islam, M. R. (2019) proposed a technique for improving the recognition accuracy of various MAFA. The MTCNN was used to tackle the occluded face identification problem. A Google Face-Net embedding model was used to extract facial traits. (SVM) was used to perform the classification task. Their experiments demonstrated that their approach performed exceptionally well on masked face identification. An AR face database, a IIIT Disguise Face Database, and an MFD database were used to evaluate their approach. To determine the best identification accuracy, multiple combinations of masked and non-masked face images were used. Table 1 displays several combinations of masked and non-masked face images used in training and testing scenarios. Table 2 illustrates the recognition accuracy of all of the Table 1 scenarios. A challenge was that the system could not identify individual faces with occlusion, affected by lighting conditions [14].

Table 1. *Masked face recognition training and testing scenario*

Dataset	Scenario	Subjects			Train			Total images Masked)	test Total (All Images(Train + Test)
		Male	Female	Total	Masked	Non-Masked	Total		
AR	Scenario 1	76	59	135	0	1666	1666	760	2426
	Scenario 2	76	59	135	704	1666	2370	935	3305
IIIT Version 1	Scenario 3	58	17	75	0	1273	1273	533	1806
	Scenario 4	58	17	75	325	148	473	208	681
Proposed MFD	Scenario 5	31	14	45	0	900	900	233	1133
	Scenario 6	31	14	45	178	895	1073	315	1388
	Scenario 7	31	14	45	0	900	900	350	1250
	Scenario 8	31	14	45	253	899	1152	461	1613

Table 2. *Masked face recognition accuracy*

Scenario	Train Image	Train Image	Train Accuracy (%)	Test Accuracy (%)
Scenario 1	Non-MAFA	Non-MAFA	100.00	90.40
Scenario 2	Non-MAFA+ MAFA	Non-MAFA	99.05	98.50
Scenario 3	Non-MAFA	Non-MAFA	100	89.49
Scenario 4	Non-MAFA+ MAFA	Non-MAFA	99.37	82.21
Scenario 5	Non-MAFA	Non-MAFA	100.00	63.52
Scenario 6	Non-MAFA+ MAFA	Non-MAFA	99.91	98.10
Scenario 7	MAFA	Non-MAFA (Complex)	100.00	47.43
Scenario 8	Non-MAFA+ MAFA (complex)	Non-MAFA (Complex)	99.05	90.24

Badrinarayanan et al (2017) present SegNet, a deep fully CNN architecture for semantic pixel-wise segmentation. It compares SegNet with other widely adopted architectures and highlights its efficiency in terms of memory and computational time during inference. The authors analyze the SegNet decoding technique and the widely used FCN to convey the practical trade-offs involved in designing segmentation architectures. The paper also evaluates the performance of SegNet on outdoor and indoor scene datasets and discusses the practical applications of the model. Additionally, it emphasizes the trade-off between memory and accuracy in segmentation architectures. The authors conclude by discussing the importance of memory and computational time during training and testing when choosing a model from a large bank of models [16].

Georgescu, M. and Ionescu, R. (2019) present an approach based on CNNs for facial expression recognition in a challenging setting with severe occlusions. The task involves recognizing the facial expression of a person wearing a VR headset, which occludes the upper part of the face. To train neural networks for this setting, the authors intentionally occlude the upper half of the face in the training examples. This approach forces the neural networks to focus on the lower part of the face and achieve better accuracy rates than models trained on entire faces. Empirical results on two benchmark data sets, FER+ and AffectNet, demonstrate that the CNN models' predictions on lower-half faces are up to 13% higher than the baseline CNN models trained on entire faces, indicating their suitability for the VR setting. Furthermore, the model's predictions on lower-half faces are no more than 10% under the baseline models' predictions on full faces, demonstrating that there are sufficient clues in the lower part of the face to accurately predict facial expressions [13].

3. System Model

3.1 Training System

The training system is the first stage of the model which comprises image acquisition, pre-processing, feature extraction/training, and testing/fine-tuning which is shown in Fig. 2 in a flow of control diagram.

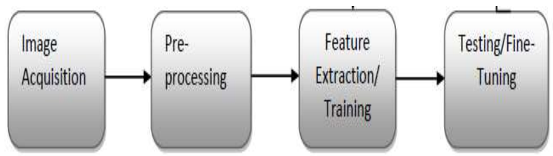


Fig. 2. *Block diagram of the Training System*

1) Image Acquisition

500 facial dataset images belonging to 100 people were scrapped from the internet and used for training.

2) Preprocessing

The images acquired were quite distorted because of uneven lighting. As a result, preprocessing steps were required before moving on to the feature extraction stage. The pre-processing carried out included two steps which were image augmentation which includes cropping, flipping, zooming, and Dataset annotation [6].

A) Image Augmentation

Image data augmentation is a technique for increasing the size of a training dataset. This is accomplished by modifying the images in the dataset. The use of image augmentation techniques introduces variances, which increases the models' capacity to generalize what they've learned to new images. The data augmentation technique uses existing data to generate additional training data. Images in the training dataset are changed into transformed versions of the same class as the original image using image data augmentation. Image augmentation aids the model in learning features such as left-to-right to top-to-bottom ordering and brightness levels in images that are invariant to transforms. Image Augmentation steps carried out are described below:

a) Random Rotation Techniques

Random Rotation technique applies a random rotation between -90 degrees and +90 degrees to the image.

b) Flipping

This is a technique that creates a mirrored version of the image. There are two types of image flips. They are the:

- **Horizontal Flip:** This is a technique that creates a mirrored image from the original one along the horizontal direction which is from left to right or vice versa.
- **Vertical Flip:** This is a technique that creates a mirrored image from the original one along the vertical direction.

c) Translation

This technique involves moving the original image in a random horizontal or vertical direction (or both). On the image sides, padding zeros are inserted.

d) Cropping

This approach is used to sample a part of the original image at random. The cropped image is large enough and includes a relevant part of the face that is significant.

B) Image Annotation

Image annotation is the process of labeling image data. Image annotation is a form of labeling. The annotated image highlights the features or characteristics that are analyzed and used to predict images. These features represent features facial images with and without occlusion. During the process of image annotation, regions such as the eyes, nose, mouth, ear, and forehead are marked. These are the exact areas of identification. Image annotation increases the accuracy by effectively training facial recognition systems.

3) Feature Extraction

Facial feature extraction is the training process where the annotated face component features are extracted. These features are the eyes, nose, mouth, ear, and forehead which are obtained from the face image. The extraction of facial features is necessary for the face recognition process to get initialized. It's crucial to be able to recognize and locate the eyes. The position of the eye is used to determine the location of all other facial features. To speed up the eye localization process, the location of the eye center is determined using an SVM classifier. The individual's face is recognized after finding the center and corners of their eye facial points [7].

4) Testing and Fine-Tuning

This is the stage after the feature extraction phase where the images are tested with a new test set. During the testing phase, the process goes back to the feature extraction phase if the robustness of the accuracy is not up to 50%, else the testing phase is successful.

3.2 Recognition System

The recognition system is the second stage of the model which comprises image acquisition, pre-processing, feature extraction, classification, and recognition. This is shown in Fig. 3 in a flow of control diagram.

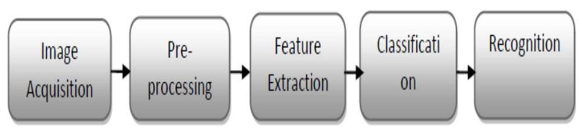


Fig. 3. *Block diagram of the Recognition System*

1. Image Acquisition

The facial dataset was acquired directly from live camera feeds of 70 persons. The images were fed into the recognition system at the image acquisition phase for the preprocessing stage. The block diagram of the recognition system is shown in Fig. 2.

2. Pre-Processing

Image preprocessing suppresses distortions and enhances necessary features that enhance recognition. Pre-processing improves the quality of images for better analysis which gives a higher recognition. The recognition system made use of automatic edge detection. During the pre-processing phase, some channels of the images were suppressed which changed the images to grayscale. This made the recognition automatically make use of a canny edge detector. Edge detection of images is a technique for image preprocessing that finds the boundaries of objects within an image. Canny Edge detection is described in detail below:

3.2.1 Canny Edge Detection

Canny edge detection detects edges within wider ranges in all directions. Canny edge detection is a Gaussian-based technique that finds a wide range of edges by detecting zero crossings in the second-order derivatives of an image [12]. Canny edge detection is a pre-processing tool used to detect edges in a very robust manner. Edges are found by separating noise from an image and extracting information from the image without disturbing its features. Low error rates, well-localized edge points, and one response per edge are all factors to consider while improving canny edge detection. The algorithm steps for canny edge detection are described below;

a) Noise Reduction/Smoothing:

Noise reduction and smoothing are accomplished by convolving the image with a Gaussian filter to remove noise or blurring the image with a Gaussian blur. To implement the canny edge detection algorithm, some procedures must be followed. The image convolution technique is applied with a Gaussian Kernel which is the 3*3, 5*5, or 7*7. An appropriate mask is taken to smoothen the image. When the mask width is greater than the detection sensitivity, the noise is reduced [8].

b) Finding Gradient

After the noise has been eliminated, the gradient of the image is used to determine the edge strength. The Gradient calculates the gradient of the image using edge detection operators to determine the edge intensity and direction. The edge corresponds to changes in the intensity of pixels which is detected by applying filters that highlight the intensity changes in both directions. These directions are called horizontal(x) and vertical(y). Sobel kernel mask is applied in both horizontal and vertical directions which uses a 3×3 convolution mask that approximates the x-direction gradient and y-direction gradient. Below is an original image and the image after finding the gradient. Some of the edges are thick and others are thin. The next step which is NMS mitigates the thick edges.

c) NMS

NMS is applied over the image to find every maxima in the gradient direction. This preserves the edges and deletes everything else. Thin edges are finally provided in the output image because NMS thins out the edges. In the NMS image below, there are still some variations in the edge intensity, some pixels seem brighter than others. This shortcoming is addressed using double thresholding and edge tracking by hysteresis [9] [10].

d) Double Threshold

The potential edges are determined using this type of threshold. Strong pixels, weak pixels, and non-relevant pixels are identified using a double threshold. The strong pixels add to the final edge because they have a very high intensity. The weak pixels are those with intensity values that aren't quite strong enough to be termed strong, but not quite small enough to be considered irrelevant. There are also other pixels which are too small and are considered non-relevant. Double threshold has two levels which are high threshold and low threshold. When the value of the edge pixel is larger than or equal to the high threshold, the threshold is marked as strong, and when it is less than or equal to the low threshold, it is marked as weak. All other pixels are tagged as weak. Hysteresis is used to identify strong thresholds and non-relevant thresholds [9] [10].

e) Edge Tracking using Hysteresis

Only if at least one of the pixels around the one being processed is strong does the hysteresis operation turn weak pixels into strong pixels.

3. Feature Extraction

This research makes use of the feature extraction technique called the Auto Encoder model. This model provides an algorithm for performing face recognition on occluded images which is based on an auto-encoder model. The model for face recognition identifies an individual based on encoded parts of the face. Images are identified by comparing a captured image with stored images of an individual in the database. The Auto Encoder Model was tested on a test set of 70 faces. The auto-encoder model carried out the following operations which mapped out various points on the faces Jawline Points, Right Eyebrow Points, Left Eyebrow Points, Nose Bridge Points, Lower Nose Points, Right Eye Points, Left Eye Points, Mouth Outline Points and Mouth Inner Points. The Auto Encoder model had an average match rate of 61.67% for faces without nose masks and 56.86% for faces with nose masks. The following operations were carried out by the Auto Encoder:

- Draws the shape using lines to connect the different points.
- Draws the shape using points and position for every landmark filtering by points parameter.
- Draws the shape using points and positions for every landmark.
- Draws the shape using points for every landmark filtering by points parameter.

Feature extraction cropped out the accurate features which was the upper region of the face which included the eye and the forehead. After cropping, the upper region is demarcated from the lower region of the face. The lower region is encoded by predicting the jawline. The Auto Encoder is used for dimensionality reduction. The face region of the test face is encoded and compared with stored faces until identification occurs, this process is carried out by the auto encoder generating a code that matches a face and comparing it with other codes used in the system to identify which one matches.

4. Classification

This stage classifies the images before recognition. Once the accuracy is less than 50%, the recognition system goes back to the feature extraction phase until an accuracy of a minimum of 60% is achieved.

4. Proposed Approach

The Auto Encoder model system is implemented using the following modules; face acquisition, face recognition without mask, and face recognition with mask and cosine difference generation. Images of persons were collected in real-time and saved to the database. The system captures the test face in real time using Python Programming Language with Microsoft Visual Studio interface. During the live image capture, the facial features are captured; the face is resized automatically and matched against the image in the database. The live capture for testing is done twice: one with a nose mask and the second without a nose mask. Matching is done and the cosine difference is generated for each image which is used to generate the percentage similarity. Table 3 below shows the cosine differences and percentage similarity for the test set of 70 images which were carried out in real time.

Table 3. *Cosine distance and percentage similarity for 70 live face image captures*

Person	Cosine Distance		Percentage Similarity	
	With Nose Mask	Without Nose Mask	With Nose Mask	Without Nose Mask
1	0.39	0.31	61	69
2	0.45	0.40	55	60
3	0.45	0.39	55	61
4	0.35	0.31	65	69
5	0.37	0.32	63	68
6	0.47	0.40	53	60
7	0.43	0.40	57	60
8	0.46	0.39	54	61
9	0.42	0.37	58	63
10	0.52	0.43	48	57
11	0.39	0.32	61	68
12	0.36	0.30	64	70
13	0.49	0.41	51	59
14	0.47	0.45	53	55
15	0.39	0.35	61	65
16	0.42	0.39	58	61
17	0.50	0.42	50	58
18	0.41	0.37	59	63
19	0.39	0.36	61	64
20	0.45	0.40	55	60
21	0.41	0.38	59	62
22	0.40	0.38	60	62
23	0.43	0.41	57	59
24	0.39	0.21	61	79
25	0.40	0.33	60	67
26	0.39	0.33	61	67
27	0.45	0.38	55	62
28	0.42	0.37	58	63
29	0.40	0.33	60	67
30	0.51	0.54	49	46
31	0.48	0.40	52	60
32	0.49	0.40	51	60
33	0.35	0.29	65	71
34	0.53	0.49	47	51
35	0.57	0.46	43	54
36	0.46	0.38	54	62
37	0.57	0.57	43	43
38	0.46	0.39	54	61
39	0.45	0.35	55	65
40	0.45	0.40	55	60
41	0.53	0.44	47	56
42	0.43	0.38	57	62
43	0.49	0.45	51	55
44	0.45	0.42	55	58
45	0.39	0.33	61	67
46	0.43	0.41	57	59
47	0.44	0.41	56	59
48	0.47	0.37	53	63
49	0.41	0.39	59	61
50	0.37	0.35	63	65
51	0.41	0.39	59	61

52	0.40	0.36	60	64
53	0.40	0.37	60	63
54	0.41	0.39	59	61
55	0.42	0.38	58	62
56	0.42	0.37	58	63
57	0.38	0.37	62	63
58	0.40	0.39	60	61
59	0.38	0.37	62	63
60	0.41	0.43	59	57
61	0.35	0.33	65	67
62	0.42	0.40	58	60
63	0.41	0.40	59	60
64	0.39	0.35	61	65
65	0.45	0.40	55	60
66	0.42	0.34	58	66
67	0.50	0.41	50	51
68	0.42	0.39	58	61
69	0.42	0.37	58	63
70	0.39	0.32	61	68

5. Performance Evaluation

The results of the experiment showed the performance of the Auto Encoder model for the facial occlusion recognition system. The system used for the implementation has a core i3 processor, 8th generation with 2.2GHz and 6GB RAM. The experiment was carried out in PYTHON using Keras, Tensorflow, and Mobilenet Libraries making use of an internal database that was used as storage. An infinix mobile phone with a 12 MP camera was used to capture live images which were stored in the database and used to test the model. The model testing was carried out on an ACER Aspire E15 system with a 720p webcam.
An Image has a numerical representation of “1”. The cosine difference is in a ratio of 0 to 1.

Cosine percentage difference = Cosine difference * 100
Percentage similarity = 100% - Cosine percentage difference.

The Cosine difference is illustrated in the image result below in Fig. 4 which had a cosine difference of 0.40 for researchers’ images without nose masks and 0.45 for researchers’ images with nose mask in Fig. 5. The FAR (Imposter) and FRR (Genuine Individual) are also presented in this section.

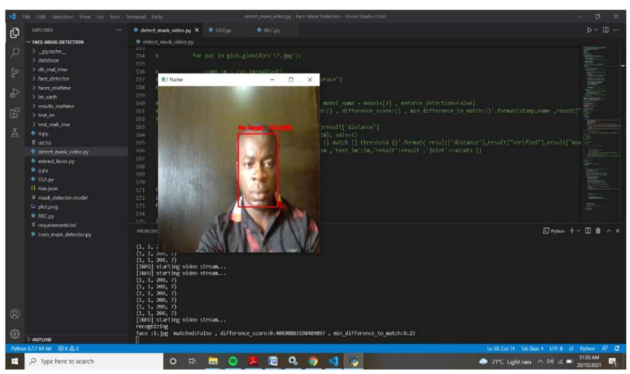


Fig. 4. *Cosine Difference Result without Nose Mask*

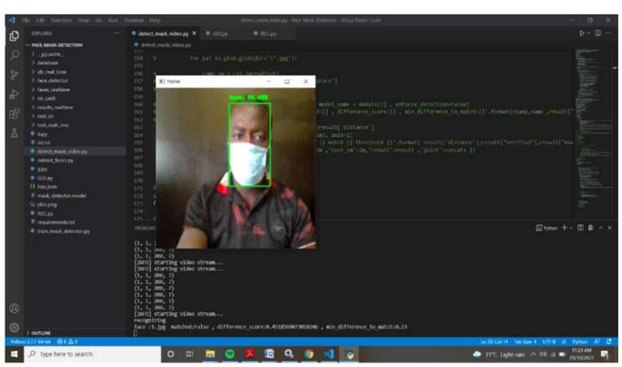


Fig. 5. *Cosine Difference Result with Nose Mask*

a) False (Imposter) Acceptance Rate [FAR]

$$\text{FAR} = \frac{NFFA}{TF} * 100$$

Where
NFFA = Number of false faces accepted
TF = Total Number of faces.

A total of 20 false or imposter faces were used but each had a very high cosine difference which meant that there was a low percentage similarity. This indicated that there were no false acceptances.

FAR Results without Nose Mask

$$\text{FAR} = \frac{0}{70} * 100 = 0\%$$

FAR Results with Nose Mask

$$\text{FAR} = \frac{0}{70} * 100 = 0\%$$

b) False (Genuine Individual) Rejection Rate [FRR]

$$\text{FRR} = \frac{NGFR}{TF} * 100$$

Where
NGFR = Number of genuine faces rejected
TF = Total number of faces

A total of 70 test faces were collected and used for training. The training was done in real-time and the result was divided into results for "with nose mask" and results for "without nose mask".

FRR Results without Nose Mask
NGFR = 2
TF = 70

$$\text{FRR} = \frac{2}{70} * 100 = 2.9\%$$

FRR Results with Nose Mask
NGFR = 6
TF = 70

$$\text{FRR} = \frac{6}{70} * 100 = 8.6\%$$

6. Contribution to Research

The research presents several useful insights and implications. It addresses the challenges of recognizing occluded faces due to nose masks and poor lighting conditions. The following insights and implications are drawn from this research.

1) Facial Occlusion Recognition System

This research introduces a facial occlusion recognition system algorithm that is divided into the train and recognition system. The train system consists of a train set of 500 images belonging to 100 people, and the recognition system involves image acquisition, preprocessing, feature extraction, classification, and recognition.

2) Auto Encoder CNN

The research utilizes an Auto Encoder CNN for feature extraction, which was implemented to recognize faces with and without nose masks in real-time. The auto-encoder achieved an average match rate of 61.67% for faces without nose masks and 56.86% for faces with nose masks. This suggests that facial recognition systems can overcome the challenge of face occlusions caused by nose masks.

3) Potential for Expansion

The research suggests that the proposed system could be expanded to incorporate iris recognition for higher robustness, allowing for the recognition of individual faces affected by full facial occlusion caused by wearing full-face masks. This indicates the potential for further development and application of the system in various biometric and computer vision domains.

4) Relevance to Biometric Applications

This research highlights the potential application of the proposed system to a variety of biometric applications, computer vision, machine learning, and automatic face recognition. This suggests that the research has implications beyond facial recognition, extending its relevance to broader domains and technologies.

5) Training System

The research outlines the training system, which includes image acquisition, preprocessing, feature extraction/training, and testing/fine-tuning. This provides a comprehensive framework for developing and training facial occlusion recognition systems.

6) Potential for Real-Time Applications

The research focuses on developing a real-time auto-encoder facial occlusion recognition system framework, indicating its potential for real-time applications in various scenarios, including security, surveillance, and access control systems. These insights and implications demonstrate the significance of the research in addressing the challenges associated with recognizing occluded faces and the potential for advancing facial recognition systems to overcome these challenges.

7. Conclusion

This paper presents a model that provides an algorithm for performing face recognition on occluded images which is based on an auto-encoder model. The model for face recognition identifies an individual based on encoded parts of the face. Images are identified by comparing a captured image with stored images of an individual in the database. It has also been established that an Encoder CNN is a good technique for recognizing images with a much higher recognition rate dealing with and without occlusions and with minimum effect from lighting conditions.

The percentage similarity of 61.67% for faces without nose mask and 56.86% for faces with nose mask which occluded the lower region of the faces was compared to [11] and [12]. It was observed that [11] detected faces and nose masks separately with a similarity percentage of 73% but could not identify the identities of faces with nose masks. [12] also detected faces and masks separately at 79.24% but could also not identify faces affected by nose mask occlusion. Real-Time Auto Encoder Facial Occlusion Recognition System Framework has proven that facial recognition systems can overcome the challenge of face occlusions caused by nose mask occlusion which is a challenge from previous research. However, this research could be widened to incorporate Iris recognition for higher robustness which will allow for recognition of individual faces affected by full facial occlusion caused by wearing a full face mask.

Compliance With Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

References

[1] R. Nandhini, N. Duraimurugan, and S. P. Chokkalingam, "Face recognition based attendance system", International Journal of Engineering and Advanced Technology, vol. 8:3-5, 2019.
[2] M.-I. Georgescu, and R. Tudor Ionescu., "Recognizing Facial Expressions of Occluded Faces using Convolutional Neural Networks", University of Bucharest, 14 Academiei Street, Bucharest, Romania, 2019.
[3] E. Ryumina, D. Ryumin, D. Ivanko, and A. Karpov, "A novel method for protective face mask detection using convolutional neural networks and image histograms", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. 44, pp.177-182, 2021.
[4] S. Khan, M. H. Javed, E. Ahmed, S. A. Shah, and S. U. Ali, "Facial Recognition using Convolutional Neural Networks and Implementation on Smart Glasses", International Conference on Information Science and Communication Technology (ICISCT), 2019.
[5] S. Alagarsamy, V. Govindaraj, M. Irfan, R. Swami, and N. M. Kumar, "Smart Recognition of Real Time Face using Convolution Neural Network (CNN) Technique", Test Engineering and Management, April 2020.
[6] S. Porcu, A. Floris, and L. Atzori, "Evaluation of Data Augmentation Techniques for Facial Expression Recognition Systems", Electronics, vol. 9, no. 11, pp. 18 – 24, 2020.
[7] S. R. Benedict, and J. S. Kumar, "Geometric shaped facial feature extraction for face recognition", International Journal of Electrical and Electronics Engineering, vol. 6, no. 9, pp. 275-278, 2016.

[8] Z. Hussain, and D. Agarwal, "A comparative analysis of edge detection techniques used in flame image processing", International Journal of Advance Research in Science and Engineering IJARSE, vol. 4, pp. 102-105, 2015.
[9] P. Gupta, and V. Gaidhane, "A new approach for flame image edges detection", Recent Advances and Innovations in Engineering, vol. 7, no. 8, pp. 1-6, 2014.
[10] K. Jayaram, G. Ramachandran, L. Vasanth, and R. Nagarajan, "Adaptive Edge Detection Algorithm for Flame and Fire Images", International Journal of Advance Research in Science And Engineering (IJARSE), vol. 5, no. 3, pp. 11 – 23, 2014.
[11] M. S. Md Suhaimin, M. H. A. Hijazi, C. S. Kheau, C. Kim On., "Real-time mask detection and face recognition using eigenfaces and local binary pattern histogram for attendance system", Bulletin of Electrical Engineering and Informatics, vol. 2, no. 4, pp. 1105-1113, 2021.
[12] S. Sen, and K. Sawant., "Face mask detection for covid_19 pandemic using pytorch in deep learning", *IOP* Conf. Series:Materials Science and Engineering, 2021.
[13] M. Georgescu, and R. Ionescu, "Recognizing Facial Expressions of Occluded Faces using Convolutional Neural Networks", University of Bucharest. vol. 6, pp, 2019.
[14] M. S. Ejaz, and M. R. Islam, "Proceeding of Masked Face Recognition using Convolutional Neural Network.," 34th Annual Conference on Sustainable Technologies for Industry 4.0 (STI) held at Dhaka, Bangladesh from 24th – 25th December 2019, 40p.
[15] D. Jaswal, V. Sowmya, and K. P. Soman, "Image classification using convolutional neural networks", International Journal of Scientific and Engineering Research, vol. 5, no. 6, pp. 2229 – 5518, 2014.
[16] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation", Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, no. 12, pp. 2481-2495, 2017.