Journal of Al-Qadisiyah for Computer Science and Mathematics Vol.17.(4) 2025,pp.Comp 121–132

Journal Of Al-Qadisiyah For Computer Science And Mathematics
JQCM
University Of AL-Qadisiyah

Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

An intelligent system for detecting and recognizing human faces

Nawar Nihad Hasobe[1], Zaid Othman Ahmed [2], Anwar Hasan Mahdi [3]

[1] Department of Computer Science, College of Science, Al-Mustansiriyah University, Al-Waziriyah, Baghdad, Iraq,.Email: nono_nini38@uomustansiriya.edu.iq
[2] Department of Computer Science, College of Science, Al-Mustansiriyah University, Al-Waziriyah, Baghdad, Iraq.Email: zied_othman@uomustansiriyah.edu.iq
[3] Department of Computer Science, College of Science, Al-Mustansiriyah University, Al-Waziriyah, Baghdad, Iraq.Email: anwar.h.m@uomustansiriyah.edu.iq

ARTICLE INFO

Article history:
Received: 09/08/2025
Rrevised form: 19/09/2025
Accepted : 21/09/2025
Available online: 30/12/2025

Keywords:

Face detect, face recogotion, CNN algorithm

ABSTRACT

Detecting human faces in any image remains a challenging problem in computer vision. The use of artificial intelligence and deep learning techniques has significantly enhanced performance in the areas of face detection and analysis. The progress is supported by both standard and complex datasets, which enable models to be trained across various scenarios. The advancement of face detection technologies is being utilized across various fields, including security systems, marketing, and healthcare systems, with innovations to enhance the system's speed and accuracy. Local database (LDB) models were used for the two models: a training set that employs a convolutional neural network (CNN) and an accompanying test set which assesses its performance using a confusion matrix (CM), and three image sizes (32,64 and 128). The findings indicate that model performance improved with the use of local data and across differing image sizes, with the pinnacle being an image size of 64 with the use of an adaptive network layer, achieving a result with an accuracy of: Accuracy (ACC) = 0.94375, Precision (P) = 0.90149, True Positive Rate (TPR) = 0.8875.

https://doi.org/10.29304/jqcsm.2025.17.42543

1. introduction

In recent years, online social networks have become increasingly popular thanks to advances in technology, allowing users to easily connect with family, friends, and colleagues. This rapid spread has made these networks a target for attackers seeking to access users' personal data. Therefore, securing the privacy of these networks is of utmost importance. Personal information spreads rapidly on these networks, putting users at risk of having their data stolen or exploited. Therefore, it is imperative to reassess the security and privacy of convolutional neural networks (CNNs) from a new perspective, particularly in light of the widespread sharing of personal content such as photos and videos online. Technologies such as neural network-powered facial recognition contribute to improved security and the provision of advanced services, but they also raise growing concerns about privacy violations and the potential for data misuse. With the increasing reliance on these technologies in various fields, protecting user data and ensuring its safe and ethical use becomes paramount [1]. Object detection and recognition an essential parts of fully understanding an image, and are a key foundation in modern computer vision techniques [2]. A Convolutional Neural Network (CNN) is a class of deep neural networks that builds upon Artificial Neural Networks (ANNs), which are primarily used for visual imagery analysis. ANN mimics and attempts to simplify the neural network architecture of the human brain. As a result, ANN shares several basic characteristics with the brain. CNN, on the other hand, was created as an improvement on ANN to meet the needs in the field of image processing when ANNs are unable to provide precise results [3]. There are techniques used in facial recognition such as cascade [4],

∗Corresponding author : Nawar Nihad Hasobe Salh

Email addresses: nono_nini38@uomustansiriyah.edu.iq

Communicated by 'sub etitor'

Deep Neural Networks [5], Landmark Detection [6], there are important applications that have been addressed in various fields, The social media ID Discovery Using Facial Recognition system [7], deep fake [8], Social Media Forensics [9], Deep fakes for Social Media Anonymization [10]. Image object detection has long been a significate challenge in the field of computer vision, necessitating detailed and complex analysis of images and understanding of their contents. This problem is central to many applications, including autonomous driving and medical image analysis. Over the years, researchers have developed new algorithms and techniques to enhance the accuracy and efficiency of object detection. Research efforts in this field continue to find more advanced and faster solutions. This system aims to develop a robust and accurate solution based on convolutional neural networks (CNNs) for effective face detection and recognition. The research focuses on developing an advanced approach to face recognition using artificial intelligence techniques, enhancing the performance of applications in areas such as security and identification. It also aims to enhance the system's accuracy to operate efficiently in diverse environments and under varying lighting conditions. This paper contributes to a framework for utilizing a local, diverse-sized database with adaptive neural network layers to enhance the accuracy and efficiency of face detection systems, thereby making a significant contribution to the existing literature in this field.

1.1 Related work:

Aleksa Ćorović et al. (2018), detail the implementation of the YOLOv3 algorithm for real-time identification of traffic participants. using the COCO dataset. Using an epoch 120, the results were precision: 0.63 recall: 0.55. [11]. Suleman Khan et al. (2019), Used smart glasses to recognize faces with the help of a CNN algorithm. The transfer learning of a trained CNN model, specifically AlexNet, was employed for face recognition, achieving an accuracy of 98.5%. [12]. Pinal Salot, (2021), discussed the CNN family, the YOLO family, and the SOLO family, where CNN works on identifying accurate squares, the YOLO family works on solving the problem in a single frame scan, and the SOLO family works on direct segmentation of instances. [13]. Muhammad Zamir and et al. (2022), using Raspberry Pi and CNN (Convolutional Neural Network) for facial recognition, accuracy achieved for the VMU, face recognition, and 14 celebrity datasets is 98%, 98.24%, 89.39%, and 95.71%, respectively. [14]. Dilnoza Mamieva et al. (2023), propose a Retina net baseline, a single-stage face detector, using the WIDER FACE and FDDB Datasets the proposed method achieves an AP of 41.0 at a speed of 11.8 FPS. [15] Heba Hamid Ali and others presented a study on developing a face recognition system applicable to Internet of Things devices, leveraging EfficientNet-B4 technology. The study employed transfer learning and fine-tuning to address the challenge of small data sizes and thereby leverage the high accuracy provided by EfficientNet-B4. After training and saving the model, it was converted to a lightweight version suitable for resource-limited devices. It was then deployed on a Raspberry Pi for use in facial recognition technology, where it achieving 97% accuracy [16]. Beibut Amirgaliyev et al. (2025), Researchers analyzed the latest developments in facial recognition, tracking, identification, and person detection technologies using a systematic review based on the PRISMA method. The research relied on more than 100 scientific papers, 142 of which were filtered to assess the quality of the reports and the methods and algorithms used, including machine learning and deep learning approaches for each task of person detection, tracking, and face recognition. The results demonstrated improvements in the accuracy of the models and the performance of the algorithms, while providing comprehensive statistics on the data used[17].

Table 1 - Comparison between previous studies

Author and year	Database	Algorithm	Results	Advantages	disadvantages
Aleksa Ćorović et al. 2018	COCO	YOLOv3	Precision: 0.63, Recall: 0.55	Real-time execution	Medium accuracy
Suleman Khan et al. 2019	Local data from smart glasses	CNN (AlexNet, Transfer Learning)	Accuracy: 98.5%	High accuracy	Depends on smart devices
Pinal Salot 2021	Experimental data	CNN, YOLO, SOLO	There are no numerical results.	Comprehensive coverage of algorithm families	The results are not quantitative.
Muhammad Zamir et al. 2022	VMU, Face Recognition, 14 Celebrity	CNN	Accuracy: 98%, 98.24%, 89.39%, 95.71%	High accuracy across multiple datasets	Hardware limited (Raspberry Pi)

Dilnoza Mamieva et al. 2023	WIDER FACE, FDDB	RetinaNet	AP: 41.0 , FPS :11.8	Good execution speed	Average accuracy (relatively low AP)
Heba Hamid Ali et al. 2023	Local data	EfficientNet-B4 (Transfer Learning, Fine-tuning)	Accuracy: 97%	Suitable for devices with limited resources	Model conversion required
Beibut Amirgaliyev et al. 2025	Review of diverse datasets	Machine Learning, Deep Learning	Improved model accuracy and improved algorithm performance	Comprehensive review, providing data statistics	Restrictions related to environmental conditions, camera angles, and privacy

1.2 Face recognition:

Using image processing to identify and verify a person's identity using facial recognition has been put to good use and is proven to be very useful and effective [18]. In today's world, facial recognition technology is being used and tested in different facets of public life. This includes the growing use of facial recognition and detection systems in primary and secondary schools, aimed at addressing long-standing issues such as campus safety [19]. For example, in airports, humans seamlessly carry out face recognition to cross the security gates. Automatic face recognition is a topic of significate research and development interest due to the global proliferation of inexpensive and effective desktop and embedded image processing systems as well as the growing interest in digital image processing and robust human-machine interfaces. Deep learning, has achieved remarkable advancements in face recognition over the past few years. There are several face recognition techniques and different models of CNNs, each having a distinct advantage stemming from its peculiar local weight-sharing architecture [21].

1.3 Convolutional neural networks (CNNs):

A convolutional neural network (CNN) is a specialized type of artificial neural network designed to efficiently process images and visual data. CNNs require less preprocessing than traditional methods, automatically extracting important features from images without requiring significant human intervention. Therefore, CNNs are among the most efficient and effective deep learning algorithms for understanding image content, making them ideal for tasks such as face recognition, image classification, and object detection in complex scenes. [22] Furthermore, CNNs have demonstrated exceptional performance in image classification and recognition, as well as high-accuracy segmentation and retrieval, making them a powerful tool in computer vision applications. [23] CNN's achievement has sparked interest outside of academia, prompting several major companies to adopt and continually develop it. Microsoft, Google, AT&T, NEC, and Facebook are among the leading companies that have contributed to the improvement of CNN architecture and its expansion into diverse applications, such as facial recognition, image analysis, and advanced computer vision. Benefits of Convolutional Neural Networks (CNNs): Focus on the image's key elements, such as edges and contours. It learns on its own, so you don't have to choose features by hand. speeds up analysis and minimizes data size, particularly when utilizing powerful processors (GPU). Great for picture identification; used in automobiles and facial recognition. Disadvantages of Convolutional Neural Networks (CNNs): A substantial amount of data is required for traning. In the absence of an adequate number of images, the performance may be suboptimal. Requires high-performance computational resources for expedited training. Prone to inaccuracies: Minor alterations in the image can result in erroneous outcomes. [24]

2. Proposed system:

The study examines the application of convolutional neural networks (CNNs) in human face recognition and classification, a powerful technique in computer vision. The methodology relies on an advanced design that can extract distinctive facial features through convolutional layers and nonlinear operators, enabling high accuracy in individual recognition and classification. The model is trained using a variety of face images that encompass different angles and lighting conditions to ensure reliable performance in real-world environments. This research aims to enhance the accuracy and efficiency of face recognition systems, thereby contributing to improved applications, including security, surveillance, and human-machine interaction.

Data used, software, tools, devices: This stage deals with determining the software and physical environment that was relied upon in implementing the experiments, including computer devices, software frameworks for developing neural networks, in addition to the type of data and images used in training and testing, (using (camera mobile type (I phone, Samsung), Videos of different sizes, Programming language (Matlab (R2023b)), computer laptop type Dell (core9)).

Preparing data and building a database for face recognition using CNN (FR_CNN): The mechanism for collecting data, processing it, and improving its quality is implemented to ensure optimal model performance. This stage also involves building an integrated database characterized by diversity, which meets the necessary requirements to accurately distinguish human faces (FR_CNN). (It was used a mobile phone was used to record many videos in different environments, 5 videos to cut out faces and create a dataset consisting of 4 people, each person holding 300 photos. The videos were filmed in different environments. Some were filmed inside buildings, while others were filmed in the outdoor environment. Additionally, it utilized the other 5 videos to create a test dataset for the same 4 people, with each person holding 100 photos (the size of the input images is 100 x 100). Figure 1 illustrates samples from the created dataset, which consists of 4 individuals. A refers to the name Ali, B refers to the name Ahmed, C refers to the name Mohammed, and D refers to the name Youssef.

Fig. 1- samples of the dataset (FR_CNN) used in this study

The goal was to build a dataset by capturing video clips from various environments, cutting them into frames and creating a local dataset for our work. It was trained on the dataset using the CNN algorithm (with simple layers and adaptive multi-layers). For improvement, additional layers were added to enhance the algorithm's operation, and a change in the classification process was made to find the most suitable selection.

2.1 CNN detection system

relied on applying a convolutional neural network to a dataset dedicated to classifying human faces using the CNN method. The training data was divided into a training group representing 80% of the total images used to improve the model's performance during the learning process, and a validation group, 20% to evaluate model’s accuracy durring in the training process. To ensure accurate results, the number of training epochs was set to 5 and a learning rate of 0.001 was used to adjust the speed of learning in the network. which helps achieve a balance between training speed and stability. The SGDM optimizer function was employed in the hidden layers due to its efficiency in Relu activation, which improves the learning process, and reduces the problem of gradient fading. The weights algorithm was used efficiently, which enhances the model's ability to adapt to the data. Additionally, a convolutional neural network was applied to the dataset to train the model to distinguish and classify human faces with high accuracy. Figure 2 shows the block diagram of the proposed system:

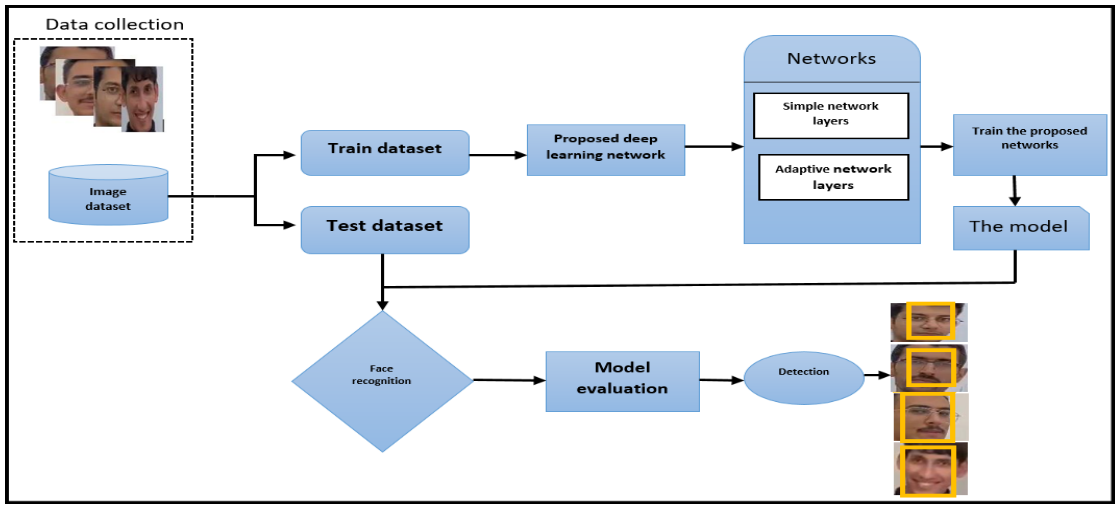
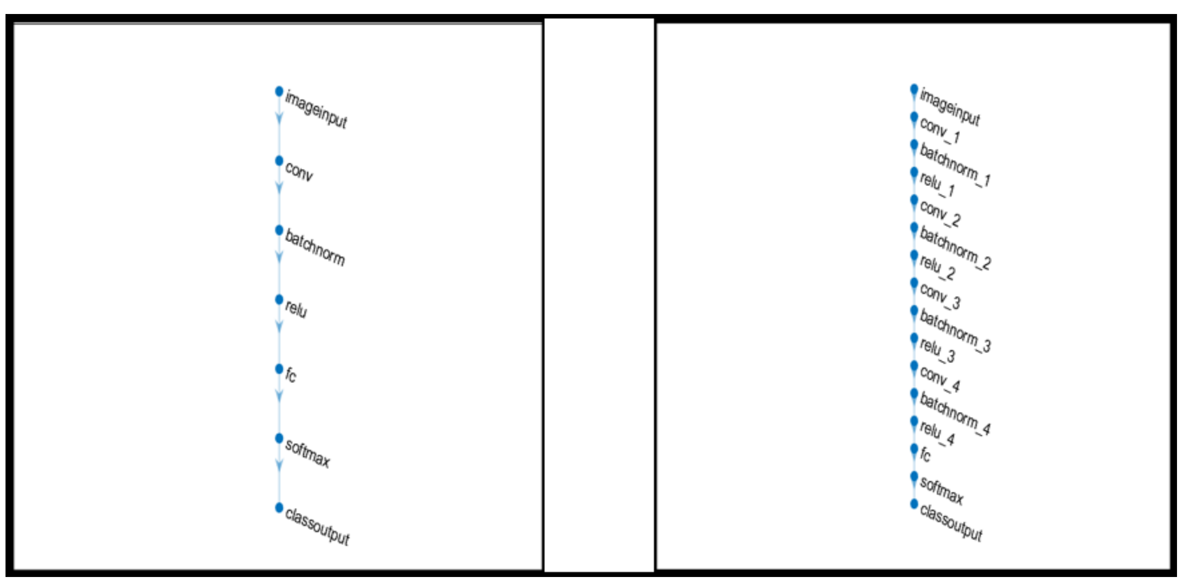


Fig. 2-The block diagram of the proposed detection system using CNN.

The training dataset with appropriate options training parameters was used to train two networks: the simple network M1 and the adaptive network M2, and their schematics are shown in Figure 3. After the training process is completed, the two network models are saved for use later in the testing phase, where these models are used to classify faces in the test dataset.

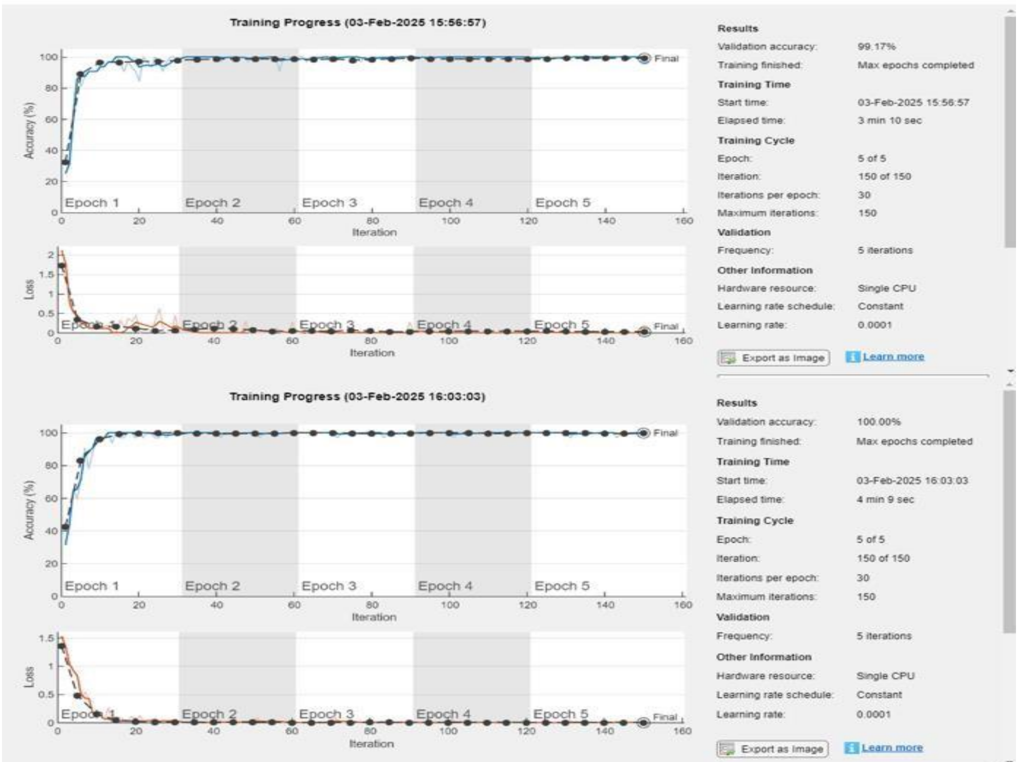


a-The Simple CNN Network Layers (M1) b-The Adaptive CNN Network Layers(M2)

Fig. 3-Network Layers.

It was noted from the previous figure that the process of improving the network by adding the layers that were reached after conducting experiments on them was the best for improvement. The improvement was done by adding layers (convolution, batchNormalization, relu, fullyConnected). Four different sizes of convolution were used (32, 64, 128), respectively.

By applying the training process, Figure 4 will show a diagram of the training process for the M1 and M2 networks that were used in this study:



(a)

(b)

a- simple Network Training (M1) b- adaptive Network Training (M2)

Fig. 4- the accuracy and loss curves for training and evaluation.

From the previous figure, it was noted that the improvement process is very clear, as the accuracy value in the simple network (M1) was 99.17%, and after improvement, it increased to 100%. This is a significant improvement.

Two models were trained to classify people's faces using two convolutional neural networks (CNNs): the first is a simple network, and the second is a modified network designed to improve performance. During the training process, a set of different parameters was studied to improve the accuracy of the models and enhance their effectiveness in distinguishing between faces. After completing the training process. The training algorithm adopted for this study is as follows:

Algorithm 1 (Image_Net_CNN)

Input : dataset

Output : Training Model

1. Initialization and Preparation
   Delete all variables and close all windows.
   Define some basic variables such as threshold (ths), number of epochs (ep), and image size (imsz).
   Define the folder containing the images.
2. Define file and folder names.
   Extract the folder name.
   Create a temporal file name to store the results.
   Create folders to save the results.
3. Load and Process Data
   Load images using imageDatastore and classify them based on folder names.
   Calculate the number of images in each class and determine the minimum number of images in each class to standardize the distribution.

Split the data into a training and test set (80% training and 20% testing).
Perform image processing and convert images to RGB if grayscale. 4. Create a Neural Network Model Define the model layers:
Input layer.
Convolution layer for feature processing.
Batch Normalization layer.
Activation layer (ReLU).
Output layer with Softmax for classification.
Final classification layer.
5. Training Setup
Specify training parameters using trainingOptions, such as:
Number of epochs (ep).
Learning rate (0.0001).
Training batches (MiniBatchSize = 32).
Early training stop function (stopTraining).
6. Model Training
Train the neural network using trainNetwork.
View training progress.
7. Evaluation and Saving Results
Predict classifications for images in the test set.
Calculate accuracy.
Save results in Excel and MAT files.
8. Early training stop function (stops training if accuracy exceeds a certain threshold (95%)).

After training the model (MDL), it is used to recognize faces in the dataset with the aim of determining the categories or patterns to which each class belongs. To evaluate the model's performance, a confusion matrix was used in addition to calculating performance indicators such as TPR, precision, and accuracy for the four classes as determined by the testing. Classifications and face detections often use accuracy as the primary means of assessment. ACC is calculated with the following equation. ACC is calculated using the equation:

$$ACC = TP + \frac{TN}{TP+TN+FP+FN} \quad \text{..... (1)}$$

Where:

- TP (True Positives): True positives.
- TN (True Negatives): True negatives.
- FP (False Positives): False positives.
- FN (False Negatives): False negatives.

True Positive Rate (TPR) or Recall is one of the measures used in assessing the performance of a model's detection. TPR is calculated with the following equation:

$$TPR = \frac{TP}{TP+FN} \quad \text{..... (2)}$$

Where:

- TP represents the number of positive cases correctly classified.
- FN represents the number of positive cases missed.

A high TPR value indicates a high ability of the model to correctly detect positive cases.

Precision is another important metric for evaluating the performance of models, especially in classification and face detection, and focuses on the correctness of the model's positive predictions. Precision is calculated using the equation:

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \text{ ..... } (3)$$

Where:

- TP (True Positives): The number of positive cases the model correctly identified.
- FP (False Positives): The number of negative cases the model incorrectly classified as positive.

The following algorithm represents this process:

Algorithm 2 (CNN-ConfusionX)

Input: Training model

Output : Confusion Matrix, Accuracy, TPR, Precision

1.Initialize and Set Variables
Delete all previous variables and close all windows.
Specify the current time to generate unique file names based on the date and time.
Load a pre-trained CNN model from a .mat file of the user's choice.
Specify the desired image size based on the loaded CNN.
Specify a folder containing images classified according to the different classes.

2.Create a folder to save the results.
Create an Rslt folder to save the experiment results.
Create a new folder within Rslt named after the used class and a timestamp.

3.Load Image Data.
Use imageDatastore to load images from the specified folder and automatically classify them based on the folder names.
Convert images to the desired size and process them using augmentedImageDatastore.

4. Predict Classifications Using a Neural Network. Classify images using cnn_net.
Calculate the overall accuracy (precision0).
Calculate and analyze the confusion matrix.
Storage the data into variables for later processing.

5.Create and Analyze the Confusion Matrix
Calculate the C matrix to measure the accuracy of the classifications.
Display and save the confusion matrix as a JPG image.
Calculate the normalized values of the C1 matrix by dividing each row by the total number of images in the class.

6.Display sample images with predictions.
Display random images with their predictions.
Print the classification information on the images (real and predicted).

7.Save the results to files.
Save the number of images, classification accuracy, and confusion matrix to a Word file.
Save the details of the classified images to an Excel file.

8.Calculate various performance metrics.
Calculate: Accuracy, Precision, TPR
Number of correct values and positive and negative values (TP, TN, FP, FN).
Save these values to a new Excel file.

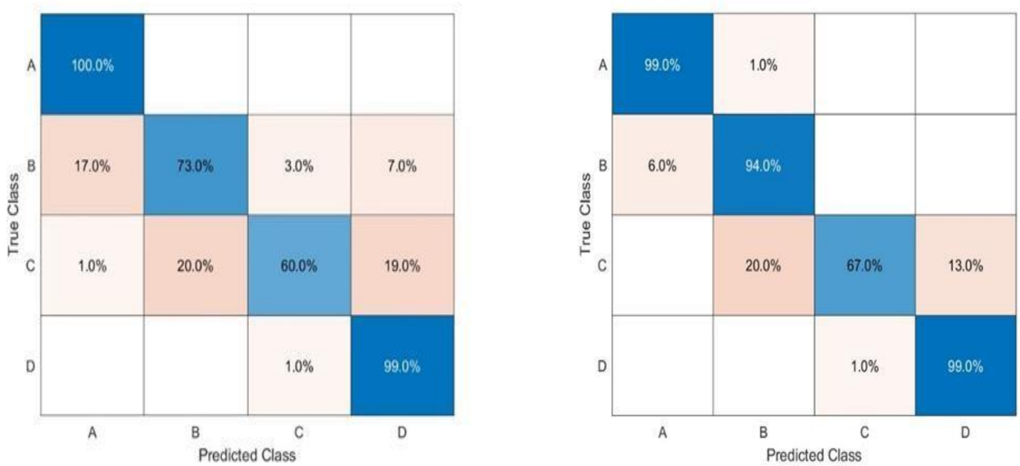
9.Save the final results to a Word file.
Store statistical values such as Precision, TPR, and Accuracy to a doc file.

10.Performance Metric Calculation Function
The metric_confusedM function takes the confusion matrix and calculates: True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN), Precision, specificity, and TPR for each class
Avoids division by zero to prevent mathematical errors.

3. Results and Discussions

The training algorithm was used to train the M1 and M2 networks, resulting in two models: MDL1_cnn and MDL2_cnn. After applying these two models to the test images in the test dataset, the results for each model were extracted. These results will be compared between the two models and their performance in classifying the four classes analyzed, with the aim of evaluating their accuracy and efficiency in recognizing different patterns within the test data.

The dataset includes 400 images representing the four classes, and the performance was evaluated using the confusion matrix, which is considered a powerful and effective tool for assessing the model's accuracy in classifying data. a confusion matrix shows the relationship between the actual ratings and the predictions made by the model for each target category. Figure 5 shows the confusion matrix of the classification process using the two models:



a-simple network layer b- adaptive network layers

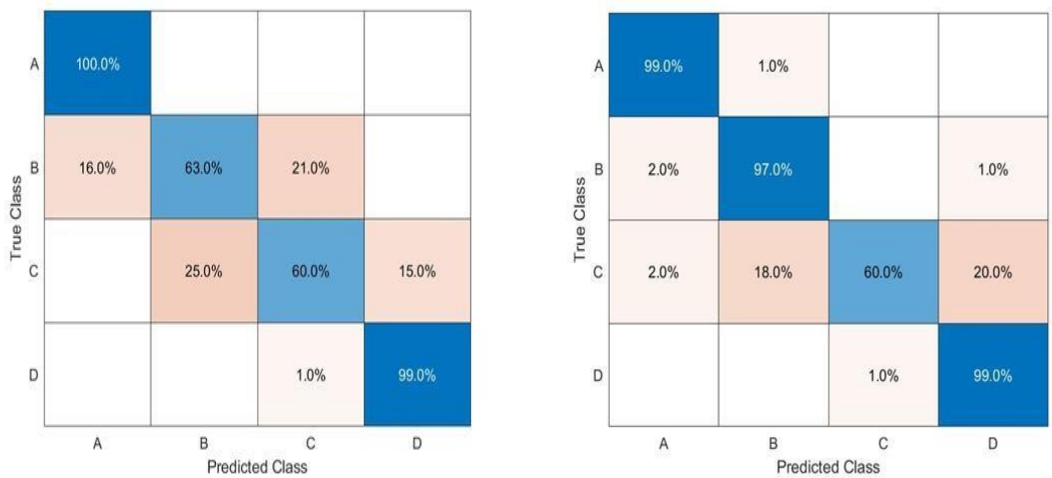
fig. 5-confusion matrix for image size (32).

Simple CNN (M1) results: The overall accuracy of the simple network was approximately 0.915 as it was able to classify most images correctly. However, there were some errors in classification, practically in cases involving low-quality images or a significate overlap between categories. The simple network recorded a classification accuracy of 0.840848, indicating that a large percentage of positive classifications were correct. The sensitivity rate (TPR) was 0.83, indicting the network's ability to correctly retrieve most positive cases. Confusion matrix: The matrix shows that the simple network achieved good performance in most categories, but it faced difficulty in distinguishing between some similar categories.

Results of the modified network (modified\enhanced) (M2) The modified network showed a significant improvement in overall accuracy, as the overall accuracy reached 0.94875, which is higher than the performance of the simple network, and the accuracy rose to 0.90737, indicating a reduction in errors in positive classifications, and the sensitivity rate (TPR) reached 0.8975, which reflects an improvement in retrieving positive cases more accurately.

Confusion matrix: The matrix shows that the modified network significantly reduces errors in classification, especially in categories that were a challenge for the simple network. By comparing the results of the two networks, it is clear that the modifications made to the modified network led to a significant improvement in overall performance, as the improvements contributed to increasing the overall accuracy and reducing classification errors, making the modified network the most suitable choice for this task. Moreover, the analysis of metrics such as precision confirms that the modified network is able to provide more reliable TPR results, particularly when handling images of varying quality. However, there are still areas for improvement, especially when dealing with images of low quality or little exchange between classes.

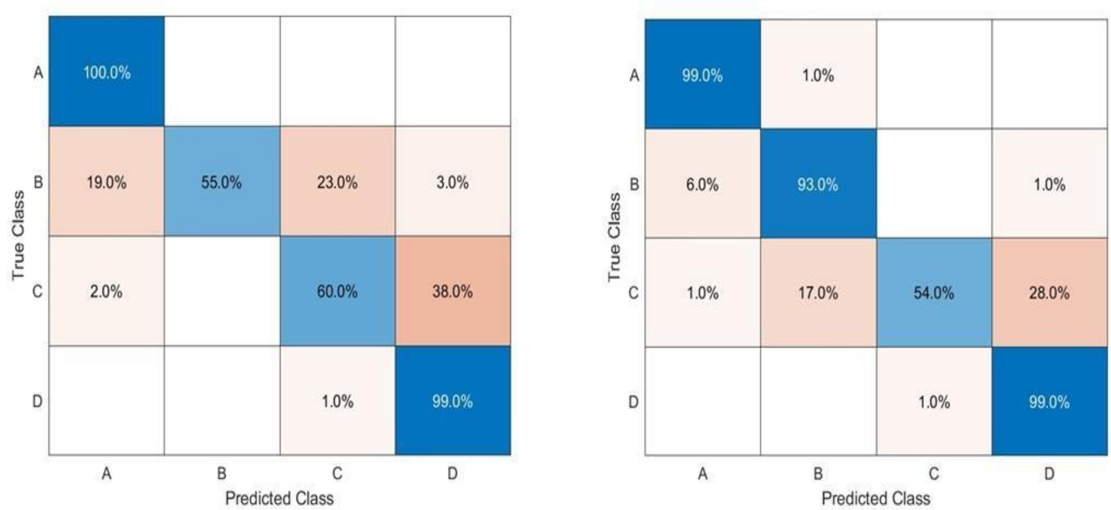
The program has been applied to different image sizes (64, 128), and the results will be displayed in sequence in figure 6 & 7.



a- Confusion Matrix in Image Size 64 for Simple Network

b- Confusion Matrix in Image Size 64 for Adaptive Network

Fig. 6- confusion matrix for image size (64).



a-Confusion Matrix in Image Size 128 for Simple Network

b- Confusion Matrix in Image Size 128 for Adaptive Network

Fig. 7- confusion matrix for image size (128).

Table 2 - citicres for image size (64, 128)

	Type layer	ACC	Precision	TPR
Image size 64	Simple layer	0.9025	0.79453	0.805
	Adaptive layer	0.94375	0.90149	0.8875
Image size 128	Simple layer	0.8925	0.81197	0.785
	Adaptive layer	0.93125	0.88176	0.8625

This table shows the criteria for images with sizes of 64 and 128. We notice from the table a noticeable difference in the results.

The detection and discrimination process in the confusion matrix is a crucial technique for identifying unknown patterns and phenomena. However, errors in discrimination can occur that leading to inaccurate results. A false detection was noted; the reason for the detection may be wrong: When the image is too small (32 pixels), details are blurry and, it is difficult to distinguish objects within it, leading to false detection. Additionally, the reason for the detection may be incorrect, if the score (classification accuracy) is less than 80%, it indicates that the model is uncertain about the result.

Table 3-Comparison between the research study and previous studies

Study and year	Algorithm	database	Image size	Performance	Notes
Aleksa Ćorović et al. 2018	YOLOv3	COCO dataset	-	Precision: 0.63, Recall: 0.55	Detect traffic participants in real time
Dilnoza Mamieva et al. 2023	RetinaNet	WIDER FACE, FDDB	-	Average Precision(AP): 41.0, Speed: 11.8 FPS	Single-stage face detection model
My study 2025	CNN adaptive	4 person each person 100 image	32	Accuracy=0.94875, Precision=0.90737, TPR=0.8975	Improved network performance for 32 image size
			64	Accuracy=0.94375, Precision=0.90149, TPR=0.8875	Improved network performance for 64 image size
			128	Accuracy=0.93125, Precision=0.88176, TPR=0.8625	Improved network performance for 128 image size

The table presents a comparison between my current work and various studies in the field of image processing and object detection. According to Ćorović and colleagues' work in 2018, the YOLOv3 COCO dataset used in real-time traffic participant detection resulted in a precision of 0.63 and a recall of 0.55. Mamieva et al. In 2023 the RetinaNet model was deployed on the WIDER FACE and FDDB databases and reporting an average precision (AP) of 41.0 and a processing speed of 11.8 frames per second. Contrarily, in my 2025 study, I trained an adaptive convolutional neural network (CNN) using a miniature dataset of four individuals (100 images per person) in varying thumbnail dimensions (32, 64, 128). Image size 32 substantially outperformed the remaining sizes (64 and 128) in terms of accuracy (0.94875), classification accuracy (0.90737), and true recall rate (0.8975). These findings confirm that the network architecture proposed in my work outperformes the preceding studies, particularly on lower- resolution images.

4.conclusion: two neural networks were used to detect faces. The first was simple, while the second was improved by adding more layers to facilitate training. The network was improved, which helped the model's accuracy. The simple network was able to perform the task, but it was not accurate enough in complex conditions. The improved network, by adding layers, was able to handle images more effectively and showed superior performance. Adding layers enhanced the model's ability to recognize fine facial details. Thus, we can see that improving the network had a significant impact on the accuracy of face detection in diverse environments. Adding layers gave the model the ability to adapt to different conditions, making it more accurate and reliable. We note that the best results were in the image size of 64, where the smaller the image size, the less accurate the detection was. And the larger the image size, the less accurate the detection was. The most important results achieved through the application of the M2 optimization network, were: ACC = 0.94375, precision = 0.90149, TPR =0.8875.

References

[1] A. K. Jain, S. R. Sahoo, and J. Kaubiyal, Online Social Networks Security and Privacy: Comprehensive Review and Analysis. (2021).
[2] J. Wang, T. Zhang, Y. Cheng, and N. Al-Nabhan, Deep Learning for Object Detection: A Survey. (2021).
[3] P. Salot, Survey on Different Object Detection and Segmentation Methods. (2021).
[4] E. S. Marquez, J. S. Hare, and M. Niranjan, Deep Cascade Learning. (2018).
[5] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K.-R. Müller, Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications. (2021).
[6] Y. Wu and Q. Ji, Facial Landmark Detection: A Literature Survey. (2018).
[7] K. J. P. Kenge, R. A. Nalawade, S. G. Nirgude, O. L. Sonawane, and D. A. Wavdhane, Social Media ID Discovery Using Facial Recognition. (2025).
[8] C. J. Zhang, A. Q. Gill, B. Liu, and M. Anwar, AI-Based Identity Fraud Detection: A Systematic Review. (2025).
[9] B. G. Bokolo and Q. Liu, Artificial Intelligence in Social Media Forensics: A Comprehensive Survey and Analysis. (2024).
[10] U. A. Çiftçi, G. Yuksek, and I. Demir, MyFace MyChoice: Privacy Enhancing Deepfakes for Social Media Anonymization. (2022).
[11] A. Ćorović, V. Ilić, S. Đurić, M. Marijan, and B. Pavković, The Real-Time Detection of Traffic Participants Using YOLO Algorithm. (2018).
[12] S. Khan, M. H. Javed, E. Ahmed, S. A. A. Shah, and S. U. Ali, Facial Recognition Using Convolutional Neural Networks and Implementation on Smart Glasses. (2019).
[13] P. Salot, Survey on Different Object Detection and Segmentation Methods. (2021).
[14] M. Zamir, N. Ali, A. Naseem, A. F. Frasteen, B. Zafar, M. Assam, M. Othman, and E.-A. Attia, Face Detection & Recognition from Images & Videos Based on CNN & Raspberry Pi. (2022).
[15] D. Mamieva, A. B. Abdusalomov, M. Mukhiddinov, and T. K. Whangbo, Improved Face Detection Method via Learning Small Faces on Hard Images Based on a Deep Learning Approach. (2022).
[16] H. H. Ali, J. R. Naif, and W. R. Humood, A New Smart Home Intruder Detection System Based on Deep Learning. (2023).
[17] B. Amirgaliyev, M. Mussabek, T. Rakhimzhanova, and A. Zhumadillayeva, A Review of Machine Learning and Deep Learning Methods for Person Detection, Tracking and Identification, and Face Recognition with Applications. (2025).
[18] M. D. Syed, A. Masood, M. Ebrahim, S. H. Adil, and K. Raza, An Improved Face Recognition Method Based on Convolutional Neural Network. J. Independent Stud. Res.–Computing (JISR-C), 22(1), Article 10, (2024). https://doi.org/10.31645/JISRC.24.22.1.10
[19] M. Andrejevic and N. Selwyn, Facial Recognition Technology in Schools: Critical Questions and Concerns. Learning, Media and Technology, 45(2), 115–128, (2020). https://doi.org/10.1080/17439884.2020.1686014
[20] K. Hamann and R. Smith, Facial Recognition Technology. Criminal Justice, vol. 34, no. 1, pp. 9–13, (2019).
[21] Y. Kortli, M. Jridi, A. Al Falou, and M. Atri, Face Recognition Systems: A Survey. Sensors, vol. 20, no. 2, p. 342, (2020).
[23] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, A Survey of the Recent Architectures of Deep Convolutional Neural Networks. Artificial Intelligence, (2020).
[24] X. Liu, Recent Progress in Semantic Image Segmentation. Artificial Intelligence, (2019).
[25] A. Hossain and M. S. A. Sajib, Classification of Image Using Convolutional Neural Network (CNN). (2019).