FACE MASK DETECTION MODEL USING CONVOLUTIONAL NEURAL NETWORK

Mamdouh M. Gomaa[1], Alaa Elnashar[1], Mahmoud M. Eelsherif[2] and Alaa M. Zaki[1]

[1] Computer Science Department, Minia University, Minia, Egypt
[2]Information Technology Department, Sinai University, North Sinai, Egypt

ABSTRACT

In current times, after the rapid expansion and spread of the COVID-19 outbreak globally, people have experienced severe disruption to their daily lives. One idea to manage the out-break is to enforce people wear a face mask in public places. Therefore, automated and efficient face detection methods are essential for such enforcement. In this paper, a face mask detection model for images has been presented which classifies the images as "with mask" and "without mask". The model is trained and evaluated using the three datasets Real-World Masked Face Dataset (RMFD), Simulated Masked Face Dataset (SMFD), and Labeled Faces in the Wild (LFW), and attained a performance accuracy rate of 99.72% for first dataset, and 100% for the second and third datasets. This work can be utilized as a digitized scanning tool in schools, hospitals, banks, and airports, and many other public or commercial locations.

KEYWORDS

COVID-19, image processing, Deep learning, Convolutional neural network (CNN).

1. INTRODUCTION

In the wake of the COVID-19 pandemic, wearing face masks has become an essential part of our daily routine. However, ensuring that individuals comply with this safety measure in public places can be challenging. To address this issue, the development of a face mask detection system using Convolutional Neural Network (CNN) has gained significant attention in recent times. CNNs are a type of deep learning neural network that are particularly effective at processing image data. By leveraging their ability to automatically learn and identify patterns in images, CNNs have been successfully applied to a wide range of computer vision tasks, including face mask detection. A face mask detection model using CNNs typically involves training a neural network on a large dataset of images that contain individuals with and without face masks. The model learns to distinguish between these two classes by analysing patterns in the images. Once trained, the model can be used to classify new images as either "with mask" or "without mask".
The implementation of face mask detection models using CNNs has numerous potential applications, from ensuring compliance with public health measures to improving workplace safety. With the ongoing global pandemic, the development of accurate and reliable face mask detection systems is more important than ever. In this paper, we presented a novel CNN model that detects whether people are wearing masks or not to reduce the spread of disease
The contributions of the proposed algorithm may be summarized as follows:

- A CNN model that is both efficient and accurate was presented, when compared to the other studied models, it obtained a promising accuracy score.
- The proposed model is lightweight. It contains five convolutional layers, five max-pooling, 458914 hyper-parameters, and one fully connected layer

- Three benchmark datasets were used in the experiments. These datasets are RMFD[11], (SMFD) [12], (LFW) [13]. It is allowed us to present accurate experiments.

The remainder of this paper is structured as follows: The related works are presented in section 2. In section 3, we will discuss a basic overview of CNN. While section 4 goes into great detail about the proposed method, section 5 presents the experimental results obtained from the proposed method and a comparison with other methods. Finally, section 6 brings the conclusion of the paper.

2. RELATED WORK

Most of the published research focuses on building the face and recognizing the identity of the face when wearing face masks. In this research, we focus on identifying whether people wear the face mask or not, to help reduce the transmission and spread of COVID 19, as scientists and researchers have proven that wearing face masks helps reduce the spread of Covid 19 infection. Bosheng Qin et al. [1] developed a facemask-wearing condition identification method that categorized facemask-wearing conditions into three categories: no facemask-wearing (NFW), incorrect facemask-wearing (IFW), and correct facemask-wearing (CFW). In face detection phase, this method achieved an accuracy of 98.70%. In a paper by Zekun Wang et al. [2], an in-browser serverless edge computing-based mask detection solution called Efficient Web-Based AI Mask Recognition (WearMask) was developed. The author employed (1) deep learning models (YOLO), (2) a high-performance neural network inference computing framework (NCNN), and (3) a stack-based virtual machine (WebAssembly), which achieved an accuracy of 89%. Sabbir et al. [3] used Principal Component Analysis (PCA) on masked and unmasked facial recognition to recognize the person. They discovered that, wearing masks significantly reduces the accuracy of facial resonance using PCA. When the detected face is veiled, the identification accuracy declines to less than 70%. Jeong-Seon Park et al. [4] also made use of PCA. The authors suggested a method for eliminating glasses from a frontal-face picture of a human. The deleted portion was rebuilt using iterative error compensation and PCA reconstruction. In Guiling Wu [5], the attention mechanism neural network is used to reduce the information loss in the subsampling process and improve the face recognition rate. After separating the masked face image by the local constrained dictionary learning method, the dilated convolution is used to reduce the resolution reduction. There were two datasets used: (1) RMFRD dataset, which had an accuracy of 98.39% in image with mask and 95.31% in image without mask, and (2) SMFRD, which had an accuracy of 98.10% in image without mask and 95.22% in image with mask. Chong Li et al. [6] used the YOLOv3. This algorithm was applied to two datasets, CelebA and WIDER FACE, for training and FDDB for testing, which achieved an accuracy of 93.9%. Nizam Ud Din et al. [7] used a GAN-based network with two discriminators, one to assist in learning the overall structure of the face and the other to focus learning on the deep missing region. The suggested model produces a full-face picture that seems natural and realistic. The CelebA dataset is used for training, and real-world images collected from the Internet are used for testing. Muhammad et al. [8] introduced MRGAN, an interactive technique. The approach relies on obtaining the microphone area from the user and rebuilding it using the Generative Adversarial Network. Deep learning real-time facial emotion categorization and identification was employed by Shaik et al. [9]. They classified seven facial expressions using VGG-16. The suggested model was trained using the KDEF dataset and attained an accuracy of 88%. A. Nieto-Rodríguez et al. [10] The authors demonstrated a method for detecting the presence or absence of an operating room mandatory medical mask. The ultimate goal is to reduce the number of false positive face detections while avoiding missing mask detections in order to activate alerts exclusively for medical personnel who do not wear surgical masks. The suggested approach achieved 95% accuracy. The following table summarize the previous models for face detection.

3. Convolutional Neural Network

CNN is a multilayer neural network structure simulating the operation mechanism of a biological vision system. This is due to its great ability to extraction important features from image. CNN has been used for image classification and image recognition problems. because it helped us to solve many of the problems we faced it in the neural network, as it provided us with local connectivity, parameter sharing (feature map) and pooling subsampling hidden unit.

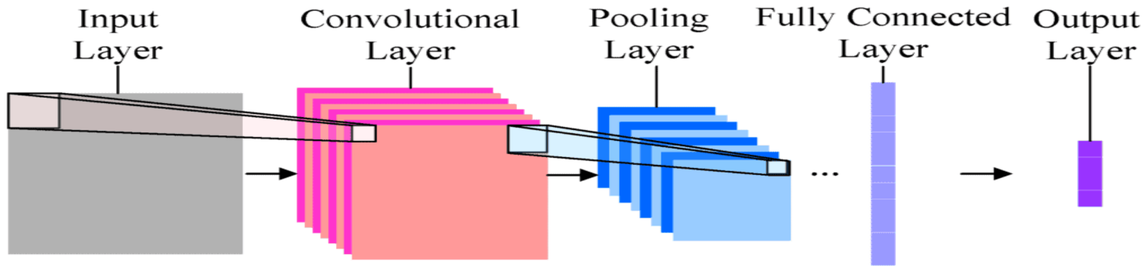

Figure 1. Structure of a convolutional neural network (CNN)[16].

As CNN includes many layers, which are convolutional layer, max pooling layer, flattening layer and full connection layer.

1) Convolutional layer: it is the activation function and a non-linear function and it has several types, the most commonly used of them
    - Relu (rectified linear unit), its importance to does not do all neural at the same time, which contributes to reducing the number of accounts performed.
    - Sigmoid which used in the output layer.

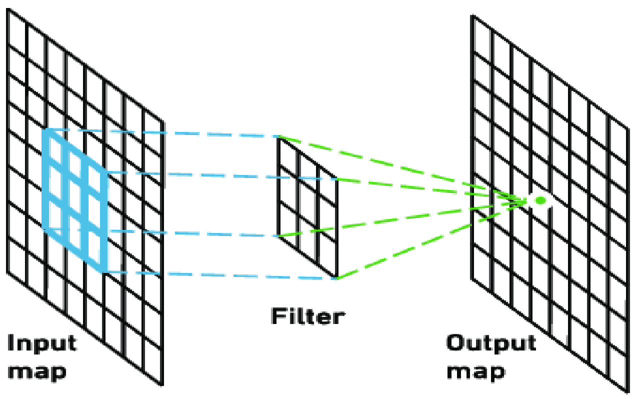

Figure 2. The Convolution Layer operation[17].

2) Max pooling layer: reduce the dimensionality of each feature map and retain the most important information of an image, spatial pooling can be different types max, average, sum.

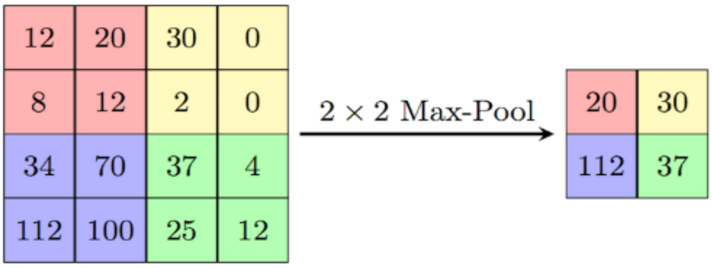

Figure 3. max pooling operation with 2 × 2 filter[18].

3) Flattening layer: we need it to convert the output of the convolutional part of CNN into one dimension feature vector to be used by Artificial neural network (ANN) part.

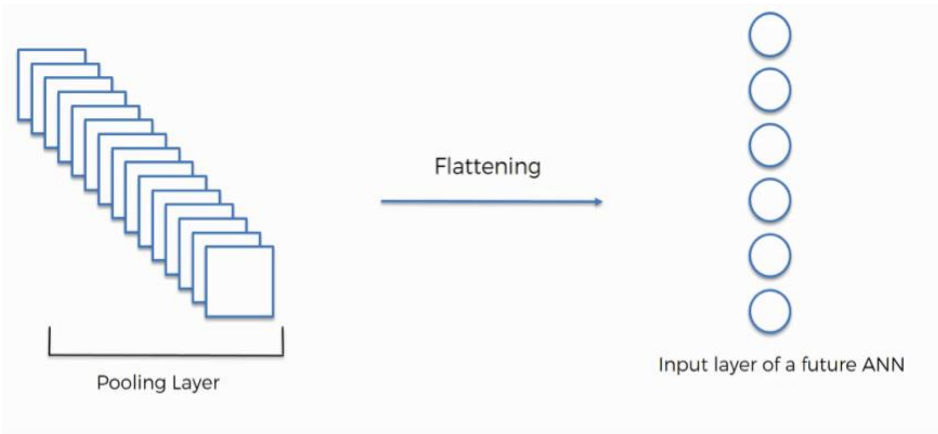

Figure 4. Flattening of Pooled Feature Maps[19].

4) Fully connected: in this phase each neuron is connected to every neuron in the previous layer, allowing the layer to learn complex non-linear relationships between the input features and the output classes. The input to each neuron in the fully connected layer is a weighted sum of the output from the previous layer, which is then passed through an activation function to introduce non-linearity.

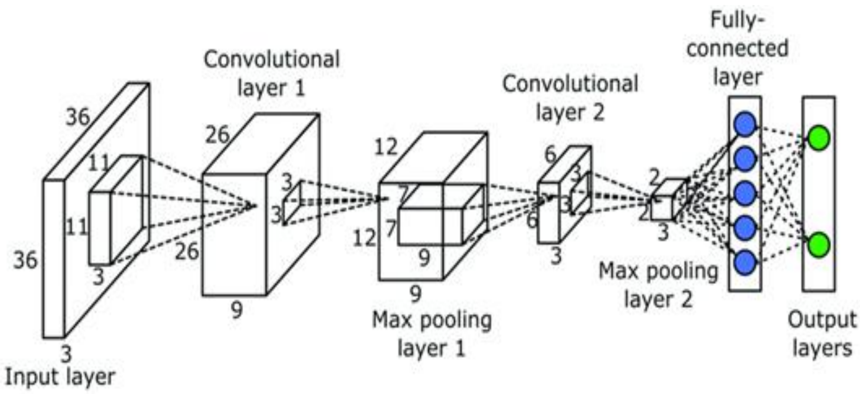

Figure 5. Convolution Neural Network Layers [20].

4. PROPOSED METHOD

In this research, we have introduced an accurate face mask detection model based on the CNN model as shown in Figure 1. The CNN approach deals with the image as a whole, while the traditional approach deals with images as a block-based algorithm. Our model consists of three stages:

1) preprocessing stage: image resized without any cropping any parts of the image for preparing it for the next stage.
2) feature extraction stage: consists of five convolution layers each of them followed by a max-pooling layer.
3) classification stage: after extracting all features, a fully connection layer connects all features with the dense layer, then the classifier categorizes the image as (masked or unmasked).

The convolutional layer used it for mining features from the input image, where each convolution layer develops its own collection of feature maps using its own set of filters (i.e., ReLU). The generated feature maps of the first convolution layer are used as input for the next max-pooling layer to generate resized pooled feature maps, and the output is considered as input for the next convolution layer. This process is repeated until the last convolution layer and max-pooling layer block. The final pooled feature maps that have been related to the final max-pooling are included as input into the global average pooling. Finally, the outputs generated from global average pooling are vectorized in a fully connected way with a dense layer. The dense layer retrieves features into two groups (masked or unmasked).

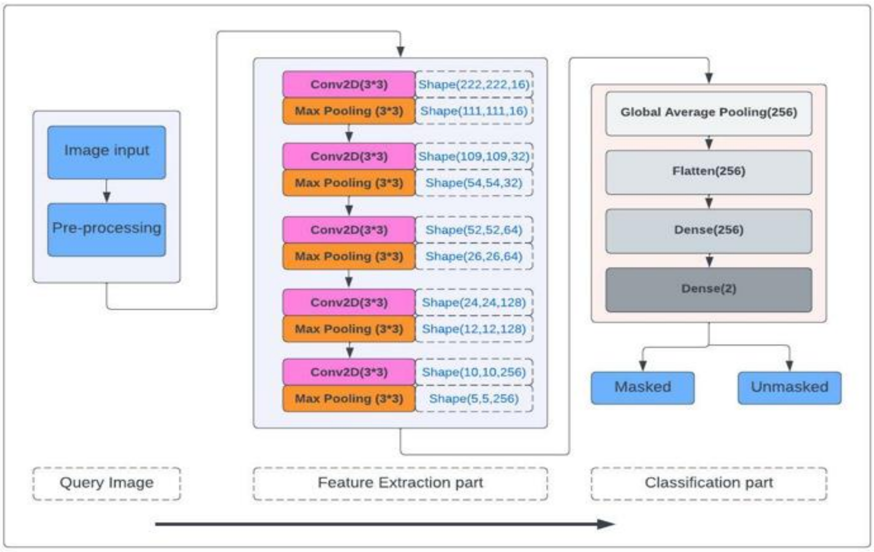

Figure 6. the proposed Convolutional Neural Network model.

5. RESULTS AND DISCUSSION

5.1. Datasets Characteristics

This model was tested on two sets of original data. The first dataset is the Real-World Masked Face Dataset (RMFD) [11], which is considered one of the largest datasets. This dataset consists of 95,000 faces, including 5,000 faces with a mask and 90,000 faces without a mask. Figure 7 shows samples of faces with and without masks. In this research, we conducted our experiments on 5,000 images of faces with a mask and 5,000 images of faces without a mask, for a total of 10,000 images. The second dataset is a Simulated Masked Face Dataset (SMFD) [12]. This set consists of 1,376 images, including 690images of masked faces and 686 images of faces without a mask. Figure 8 shows samples of faces with a mask and faces without a mask. The third dataset is Labeled Faces in the Wild (LFW) [13]. The dataset contains more than 13,000 images of faces collected from the web. Face masks have been placed on some of the dataset images. Figure 9 shows samples of faces with a mask and faces without a mask. All details of division of datasets shown in table 1, the images in the three dataset divided into three group (training group, test group, and verification group).

Table 1. the characteristics of RMFD, SMFD, LFW datasets

Dataset	**Composition**				**No. of training Images**				**No. of validation Images**				**No. of testing Images**				**The input shape**
RMFD [11]	10000 images				5000 images				2500 images				2500 images				224 × 244 pixels
	Masked	5000	unmasked	5000	masked	2500	unmasked	2500	masked	1250	unmasked	1250	masked	1250	unmasked	1250	
SMFD [12]	1376 images				688 images				344 images				344 images				224 × 244 pixels
	Masked	690	unmasked	686	masked	344	unmasked	344	masked	173	unmasked	171	masked	173	unmasked	171	
LFW [13]	13233 images				6617 images				3308 images				3308 images				224 × 244 pixels
	Masked	6616	unmasked	6617	masked	3308	unmasked	3309	masked	1654	unmasked	1654	masked	1654	unmasked	1654	

Figure 7. sample of RMFD dataset images

Figure 8. sample of SMFD dataset images

Figure 9. sample of LFW dataset images

5.1. Evaluation Metrices

To calculate the strength of the proposed model, we used the following accuracy measurement:

$$\text{Accuracy} = \frac{(T_N + T_P)}{(T_P + F_P + T_N + F_N)} \times 100 \tag{1}$$

$$\text{Precision} = \frac{(T_P)}{(T_P + F_P)} \times 100 \tag{2}$$

$$\text{Recall} = \frac{(T_P)}{(T_P + F_N)} \times 100 \tag{3}$$

$$\text{F1} - \text{score} = \frac{2 * (\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \times 100 \tag{4}$$

Table 2 shows a confusion matrix, which is used to measure the classification algorithm performance. The True Positive (TP) is the number of masked images that are truly detected as masked images. A False Positive (FP) is the number of unmasked images that are falsely detected as masked images. A False Negative (FN) is the number of masked images that are falsely detected as unmasked images. True Negative (TN) is the number of unmasked images that are truly detected as unmasked images.

Table 2. confusion matrix

		Actual Value	
		Positive	**Negative**
Predicted Value	**Positive**	**TP (True Positive)**	**FP (False Positive)**
	Negative	**FN (False Negative)**	**TN (True Negative)**

5.2. Output and Results

Our method was tested over three datasets: RMFD, SMFD, and LFW. The results are compared to techniques that have recently been published (M. Dwisnanto et al. [14], Loey et al. [15]). Table 3 displays a confusion matrix for three datasets, where sign (+) denotes masked images and sign (-) denotes unmasked images.

Table 3. The Confusion Matrix of the Proposed Approach with SMFD, RMFD, and LFW Dataset

Dataset	Classes	+	-	Total
RMFD	+	1245	5	1250
	-	2	1248	1250
	Total	1247	1253	2500
SMFD	+	173	0	173
	-	0	171	171
	Total	173	171	344
LFW	+	1654	0	1654
	-	0	1654	1654
	Total	1654	1654	3308

5.3.1. Results of RMFD dataset

Experiments were performed on RMFD dataset and the results were as follows: In epoch 15, the accuracy, F1 score, and time testing (TT) were 98.64%,.985, and 127, respectively, In epoch 30, the accuracy, F1 score, and TT were 98.96%,.988, and 134, respectively, In epoch 45, the accuracy, F1 score, and TT were 99.72%,.996, and 143, respectively, In epoch 75, the accuracy, F1 score, and TT were 99.72%,.996, and 149, respectively, In epoch 100, the accuracy, F1 score, and TT were 99.72%,.996,153 and , respectively. The optimal experiment was at epoch 45 as shown in table 4.

Table 4. the results of the proposed approach on RMFD dataset.

	TP	**FP**	**TN**	**FN**	**P**	**R**	**F1-score**	**Accuracy%**	**TT (sec.)**
Epoch 15	1229	21	1237	13	.983	.989	.985	98.64	127
Epoch 30	1233	17	1241	9	.986	.992	.988	98.96	134
Epoch 45	**1245**	**5**	**1248**	**2**	**.996**	**.998**	**.996**	**99.72**	**143**
Epoch 75	1245	5	1248	2	.996	.998	.996	99.72	149
Epoch 100	1245	5	1248	2	.996	.998	.996	99.72	153

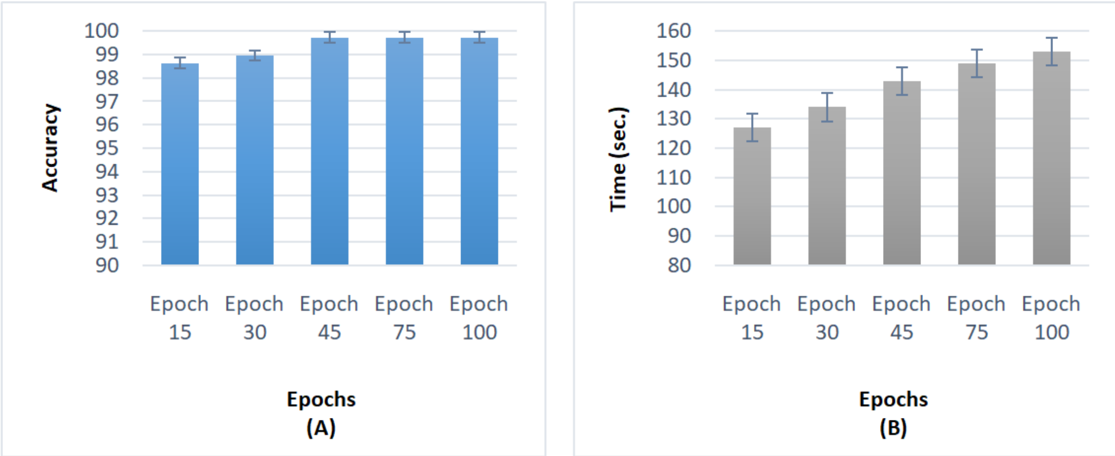

Figure 10. (A) Accuracy for each epoch for RMFD dataset, (B) Time for each epoch for RMFD dataset.

Figure 10 (A) shows the variation of the accuracy at a different number of epochs, (B) shows the variation of the time testing at a different number of epochs.

5.3.2. Results of SMFD Dataset

The results of the practical experiments on the SMFD dataset are given on a different number of epochs, In epoch 15, the accuracy, F1 score, and time testing (TT) were 97.38%,.973, and 26, respectively, In epoch 30, the accuracy, F1 score, and TT were 99.13%,.990, and 29, respectively, In epoch 45, the accuracy, F1 score, and TT were 100%,1, and 34, respectively, In epoch 75, the accuracy, F1 score, and TT were 100%,1, and 37, respectively, In epoch 100, the accuracy, F1 score, and TT were 100%,1, and 41, respectively. The optimal experiment was at epoch 45 as shown in table 5.

Table 5. the results of the proposed approach on SMFD dataset.

	TP	**FP**	**TN**	**FN**	**P**	**R**	**F1-score**	**Accuracy %**	**TT (sec.)**
Epoch 15	168	5	167	4	.971	.976	.973	97.38	26
Epoch 30	171	2	170	1	.988	.994	.990	99.13	29
Epoch 45	**173**	**0**	**171**	**0**	**1**	**1**	**1**	**100**	**34**
Epoch 75	173	0	171	0	1	1	1	100	37
Epoch 100	173	0	171	0	1	1	1	100	41

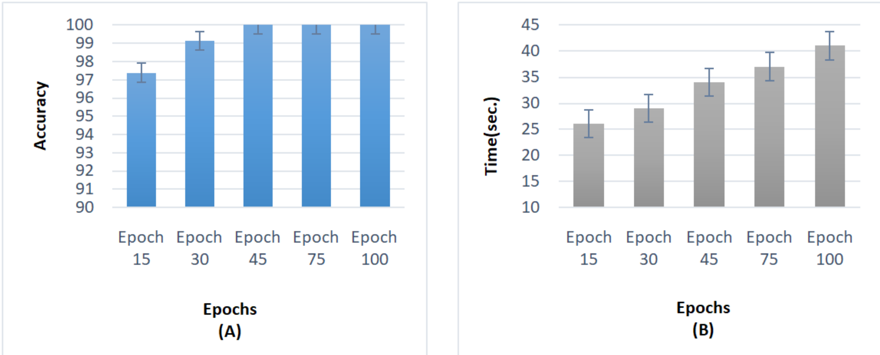

Figure 11. (A) Accuracy for each epoch for SMFD dataset, (B) Time for each epoch for SMFD dataset.

Figure 11 (A) shows the variation of the accuracy at a different number of epochs, (B) shows the variation of the time testing at a different number of epochs.

5.3.3. Results of LFW Dataset

When performing our proposed model over the LFW dataset at different epoch numbers, the practical experiment results are given as follows: In epoch 15, the accuracy, F1 score and (TT) were 99.78%, .997 and 83, respectively, In epoch 30, the accuracy, F1 score, and TT were 99.84%, .998 and 89 respectively, In epoch 45, the accuracy, F1 score and TT were 100%, 1 and 93, respectively, In epoch 75, the accuracy, F1 score and TT were 100%, 1 and 102, respectively, In epoch 100, the accuracy, F1 score, and TT were 100%, 1, and 111, respectively. The optimal experiment was at epoch 45 as shown in table 6.

Table 6. the results of the proposed approach on LFW dataset.

	TP	FP	TN	FN	P	R	F1-score	Accuracy %	TT (sec.)
Epoch 15	1650	4	1651	3	.997	.998	.997	99.78	83
Epoch 30	1651	3	1652	2	.998	.998	.998	99.84	89
Epoch 45	**1654**	**0**	**1654**	**0**	**1**	**1**	**1**	**100**	**93**
Epoch 75	1654	0	1654	0	1	1	1	100	102
Epoch 100	1654	0	1654	0	1	1	1	100	111

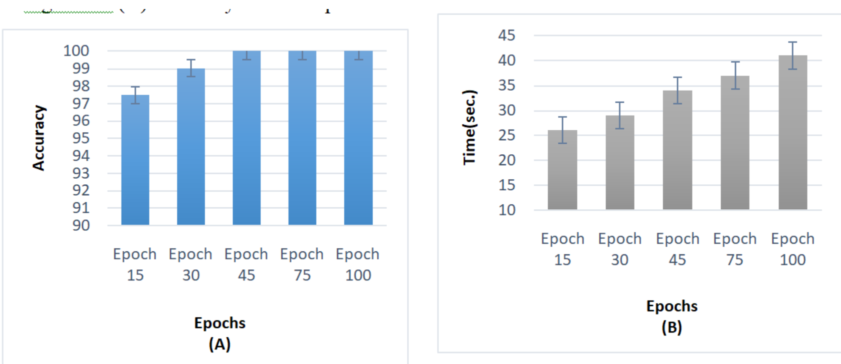

Figure 12. (A) Accuracy for each epoch for
LFW dataset, (B) Time for each epoch for LFW dataset.

Figure 12 (A) shows the variation of the accuracy at a different number of epochs, (B) shows the variation of the time testing at a different number of epochs.

5.3.4. Comparison with Related Works over RMFD Dataset

While conducting practical experiments using the proposed model on the RMFD dataset, we compared our results with the previously published results (Loey et al. [15]), and the comparison was done on a number of parameters, F1-score and accuracy, in table 7 the obtained results .967, .994, .996 and .996 for [15] [1], [15] [2], [15] [3] and proposed model respectively over F1-Score, 96.78%, 99.4%, 99.64% and 99.72% for [15] [1], [15] [2], [15] [3] and proposed model respectively over accuracy, 23,591,816 and 458,914 for [15] and proposed model respectively over number of parameters, as shown in figure 13.

Table 7. parameters, F1-score and accuracy, comparison between the proposed model and previously related work over RMFD dataset.

Method	Architecture	Parameters	F1-Score	Accuracy (%)
Loey et al. [15]	Resent50 + Decision trees [1]	23,591,816	.967	96.78
	Resent50 + SVM [2]		.994	99.4
	Resent50 + Ensemble [3]		.996	99.64
Proposed model	**CNN model**	**458,914**	**.996**	**99.72**

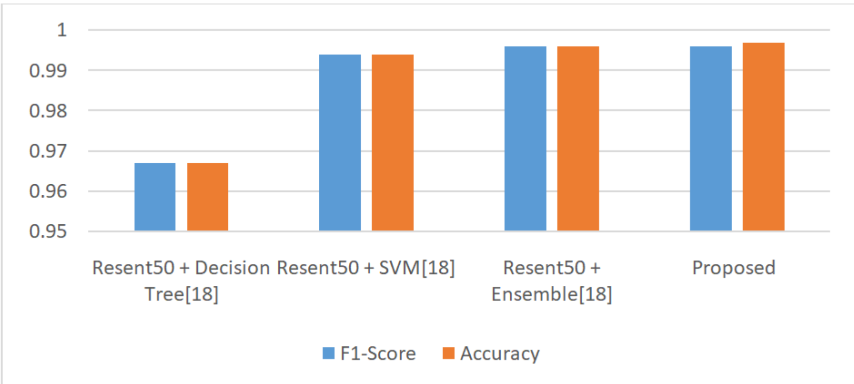

Figure 13. F1-score and accuracy comparison with Resent50 +decision trees, Resent50 +SVM, Resent50 +Ensemble and proposed model over RMFD dataset.

5.3.5. Comparison with Related works over SMFD Dataset

According to the comparison of our proposed model with the previous approaches (Loey et al. [15] and M. Dwisnanto et al. [14]) over the SMFD dataset, the comparisons were made on the number of parameters, F1-score, and accuracy. Table 8 shows the details of the results as follows .956, .994, .994 and 1 for [15] [1], [15] [2], [15] [3] and proposed model respectively over F1-Score, 95.64%, 99.49%, 99.49%,99.72% and 100% for [15] [1], [15] [2], [15] [3],[14] and proposed model respectively over accuracy, 23,591,816, 668,746 and 458,914 for [15], [14] and proposed model respectively over number of parameters, as shown in figure 14.

Table 8. parameters, F1-score and accuracy, comparison between the proposed model and previously related work over SMFD dataset.

Method	Architecture	Parameters	F1-Score	Accuracy (%)
Loey et al. [15]	Resent50 + Decision trees [1]	23,591,816	.956	95.64
	Resent50 +SVM [2]		.994	99.49
	Resent50 +Ensemble [3]		.994	99.49
M. Dwisnanto et al. [14]	CNN model	668,746	-	99.72
Proposed model	**CNN model**	458,914	**1**	100

Figure 14. F1-score and accuracy comparison with Resent50 +decision trees, Resent50 +SVM, Resent50 +Ensemble, M. Dwisnanto et al. model and proposed model over SMFD dataset.

5.3.6. Comparison with related Works over LFW Dataset

After using the proposed model in conducting practical experiments on the LFW dataset, we compared the previously published results (Loey et al. [15] and M. Dwisnanto et al. [14]), where the comparison was made in terms of the number of parameters, F1-score, and accuracy. It is clear from Table 9 that the F1-score and accuracy value are similar at [14],[15][2],[15][3] and the proposed model, but differ in [15][1] which F-score is .998 and accuracy is 99.89% , and also they differ in the number of parameters such that 23,591,816, 668,746 and 458,914 for [15], [14] and the proposed model, respectively, as shown in figure 15.

Table 9. parameters, F1-score and accuracy, comparison between the proposed model and previously related work over SMFD dataset.

Method	Architecture	Parameters	F1-Score	Accuracy (%)
Loey et al. [15]	Resent50 + Decision trees [1]	23,591,816	.998	99.89
	Resent50 +SVM [2]		1	100
	Resent50 +Ensemble [3]		1	100
M. Dwisnanto et al. [14]	CNN model	668,746	1	100
Proposed model	**CNN model**	**458,914**	**1**	**100**

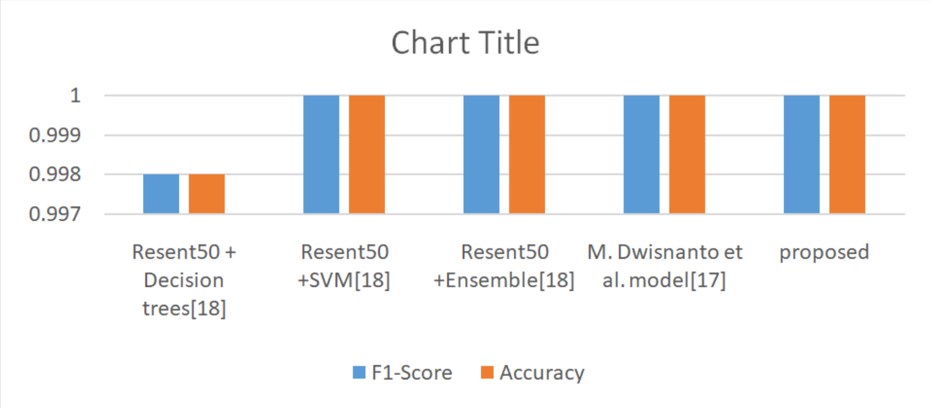

Figure 15. F1-score and accuracy comparison with Resent50 +decision trees, Resent50 +SVM, Resent50 +Ensemble, M. Dwisnanto et al. model and proposed model over LFW dataset.

6. Conclusion

In conclusion, we presented a face mask detection model for images that classifies them as "with mask" and "without mask" using a convolutional neural network (CNN). The proposed model is lightweight, efficient, and was trained and evaluated using three benchmark datasets, achieving high accuracy rates of 99.72%, 100%, and 100%, respectively. The model can be used as a digitized scanning tool in various public or commercial locations to ensure compliance with public health measures and improve workplace safety, particularly during the ongoing COVID-19 pandemic. Compared to other models, our proposed algorithm showed promising results, and the model's contributions have been summarized. In summary, this research demonstrates the potential of deep learning models to improve public health and safety by detecting face masks in real-time.

References

[1] B. QIN and D. Li, Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19, May 2020, doi: 10.21203/rs.3.rs-28668/v1.

[2] WearMask: Fast In-browser Face Mask Detection with Serverless Edge Computing for COVID-19 Zekun Wang, Pengwei Wang, Peter C. Louis, Lee E. Wheless, and Yuankai Huo, Member, IEEE

[3] M.S. Ejaz, M.R. Islam, M. Sifatullah, A. Sarker, Implementation of principal component analysis on masked and non-masked face recognition, in: 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT), 2019, pp. 1–5, https://doi.org/10.1109/ ICASERT.2019.8934543.

[4] Jeong-Seon Park, You Hwa Oh, Sang Chul Ahn, and Seong-Whan Lee, Glasses removal from facial image using recursive error compensation, IEEE Trans. Pattern Anal. Mach. Intell. 27 (5) (2005) 805–811, doi: 10.1109/TPAMI.2005.103.

[5] GuiLing Wu “Masked Face Recognition Algorithm for a Contactless Distribution Cabinet” Mathematical Problems in Engineering, Volume 2021, Article ID 5591020, 11 pages, 2021 (https://doi.org/10.1155/2021/5591020)

[6] C. Li, R. Wang, J. Li, L. Fei, Face detection based on YOLOv3, in:: Recent Trends in Intelligent Computing, Communication and Devices, Singapore, 2020, pp. 277–284, doi: 10.1007/978-981-13-9406-5_34.

[7] N. Ud Din, K. Javed, S. Bae, J. Yi, A novel GAN-based network for unmasking of masked face, IEEE Access 8 (2020) 44276–44287, https://doi.org/10.1109/ ACCESS.2020.2977386.

[8] M.K.J. Khan, N. Ud Din, S. Bae, J. Yi, Interactive removal of microphone object in facial images, Electronics 8 (10) (2019) , Art. no. 10, doi: 10.3390/ electronics8101115.

[9] S. A. Hussain, A.S.A.A. Balushi, A real time face emotion classification and recognition using deep learning model, J. Phys.: Conf. Ser. 1432 (2020) 012087, doi: 10.1088/1742-6596/1432/1/012087.

[10] A. Nieto-Rodríguez, M. Mucientes, V.M. Brea, System for medical mask detection in the operating room through facial attributes, Pattern Recogn. Image Anal. Cham (2015) 138–145, https://doi.org/10.1007/978-3-319-19390-8_16.

[11] Z. Wang, et al., Masked face recognition dataset and application, arXiv preprint arXiv:2003.09093, 2020.

[12] prajnasb, “observations,” observations. https://github.com/prajnasb/observations (accessed May 21, 2020).

[13] E. Learned-Miller, G.B. Huang, A. RoyChowdhury, H. Li, G. Hua, Labeled faces in the wild: a survey, in: M. Kawulok, M.E. Celebi, B. Smolka (Eds.), Advances in Face Detection and Facial Image Analysis, Springer International Publishing, Cham, 2016, pp. 189–248.

[14] M. Dwisnanto, Duy-Linh Nguyen, Kang-Hyun Jo, “Real-Time Multi-view Face Mask Detector on Edge Device for Supporting Service Robots in the COVID-19 Pandemic”, ResearchGate International Publishing, April 2021, (https://www.researchgate.net/publication/350619412).

[15] Loey, M. Manogaran, G., Taha, M.H.N., Khalifa, N.E.M.: A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the Covid-19 pandemic Measurement 167, 108288 (2021), (http://www.sciencedirect.com/science/article/pii/S0263224120 308289).

[16] https://www.linkedin.com/pulse/face-recognition-mobilenet-architecture-using-transfer-srishti-gupta

[17] https://medium.com/swlh/building-a-simple-convolution-layer-from-scratch-317f82dc3805

[18] https://paperswithcode.com/method/max-pooling

[19] https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-3-flattening

[20] https://mikewinters.io/2019/06/29/ai-sonification/