JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 1, Maret 2023, Pp. 281-290

FACE MASK DETECTION UNDER LOW LIGHT CONDITION USING CONVOLUTIONAL NEURAL NETWORK (CNN)

Naufal Muhammad Athif*1), Febriyanti Sthevanie2), Kurniawan Nur Ramadhan3)
1. Telkom University, Bandung, Indonesia
2. Telkom University, Bandung, Indonesia
3. Telkom University, Bandung, Indonesia

Article Info
Keywords: Face Mask Detection, Low Light, Keras, TensorFlow, PIL

Article history:
Received 19 November 2022
Revised 3 December 2022
Accepted 10 December 2022
Available online 1 March 2023

DOI :
https://doi.org/10.29100/jipi.v8i1.3324

* Corresponding author.
Corresponding Author
E-mail address:
naufalmathif@student.telkomuniversity.ac.id

ABSTRACT
The COVID-19 pandemic has been around for 3 years, and the virus is still spreading until now and using mask is an alternative for people to not get infected, but some people tend to let go of the mask for inconvenience reasons, especially under low light conditions which is difficult for humans to identify. Thus, this paper proposed and implemented a face mask detection model which can accurately detect a person that using a mask or not in such a condition as low light by using Convolutional Neural Network (CNN) architecture with OpenCV, TensorFlow and Keras. To achieve this, the first step is to transform the data by using Python Imaging Library (PIL) to create a low light image, then we process the data by using Contrast Limited Adaptive Histogram Equalization and with Gamma Correction. The second step is to augment the data by using TensorFlow ImageDataGenerator and define the CNN model. The final step is to create the face mask prediction by using Haar Cascade Algorithm to detect the face mask. The results of this research shows that CNN model can be trained with a recreational low light images to detect face mask under low light conditions. The result of the model produced an accuracy of 98%.

I. INTRODUCTION

IN 2019, COVID-19 start its infection into the world, including Indonesia. With one COVID-19 inhalation that is invisible to the naked eye, humans who do not have a strong immunity will be infected. If a person is infected, then the result from it can go up to even the worst-case scenario, which is an inevitable death. Therefore, people began to create a solution that can help others to fend off the virus. One of the alternatives that is used is a face mask. However, some of the people tend to neglect using mask within crowded places. Thus, many researches creating a face mask detection system that uses many models such as Convolutional Neural Network (CNN).

Face Mask Detection methods are spreading widely on the internet within time. The pandemic gains a curiosity towards researchers which they improve classification and performance of the face mask detection system, thus the results are getting better and better. For example, in 2017, Preeti Nagrath *et al.* [1] conduct research with the title “SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2” which uses deep learning, TensorFlow, Keras, and OpenCV to detect face mask. The approach for this paper is by a Single Shot Multibox Detector as a face detector and MobileNetV2 architecture as a framework for the classifier. The results are obtained throughout this research with an accuracy score of 0.9264 and an F1 score of 0.93. Another example such as in 2020, Arjya Das, Mohammad Wasif Ansari, and Rohini Basak [2] which they conducted research with the title “Covid-19 Face Mask Detection Using TensorFlow, Keras, and OpenCV” that can also detect a face in motion. The result for the research attains an accuracy for up to 95.77% and 94.58% which uses Sequential Convolutional Neural Network Model to detect the mask without overfitting. In 2020, Shilpa Sethi, Mamta Kathuria, and Trilok Kaushik conduct research with the title “Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread” [3] which contains ResNet50, AlexNet, and MobileNet as a baseline models followed by an ensemble of a one-stage and two-stage detectors to achieve low inference time and high accuracy. This paper also proposed a bounding box transformation to improve localization performance during mask detection. The results from the ResNet50 model achieve high accuracy of 98.2%.

From the research that is studied about the face mask detection, one of the downsides is that most of the paper does not focus on low light condition. The case for this is that studying the face mask detection under night light is a difficult problem because there are internal and external factor that the system features needed for the mask detection, especially under low light. For example, the brightness of an image that is captured under low light conditions tend to be very low and the contrast will be severely reduced [4]. Humans can easily detect the use of

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 1, Maret 2023, Pp. 281-290

the face mask under low light conditions. But it is difficult for a machine to detect the use of face mask under low light conditions. One of the other problems is that an image that is acquired using computer vision system under low light conditions tends to contain multiple characteristic such as high noise, lousy illumination, reflectance and bad contrast which makes object detection more difficult [5]. Thus, A face mask detection system needs a special approach to overcome low light conditions. Using some low-illumination tools to help identify the mask, but using these tools is not easily implemented because of its high cost. One of the most popular approaches of face recognition research is created through deep learning, which is data preparation dependent as to performance and accuracy. We can use deep learning method to create a face mask detection system.

Thus, this paper proposed an approach on face mask detection under low light condition which is efficient and can be accurate in terms of detection under low illumination images, and can be implemented in real-time. Using pre-trained model is achievable, but this paper's intention is to create a light-weight CNN model so that the people can use this method. This paper focuses on face mask detection system that can detect face mask under low light conditions and the approach for this study gains an accuracy that is comparable to the existing face mask detection method. To do this, we collect a public face mask dataset combined with our own. CNN architecture is used to train the model to detect the face mask along with some Machine Learning (ML) packages such as TensorFlow, Keras, and OpenCV along with Python Imaging Library (PIL) for the low light manipulation. This approach has its pros which is easy to implement with a low cost and is able to achieved a promising result.

The contributions that can be considered in this study is that this approach is one of the alternatives to achieve a face mask detection system under a low light condition. Using PIL as a low light manipulation for the dataset can give a resemblance of a real low light image that can be used in many cases which includes a low light image detection.

The organization for the rest of this paper is as followed. Section 2 reviews the related previous work. Section 3 explain about the approach for the face mask detection. Section 4 evaluate and analyze results. Section 5 conclusion and future work.

II. Proposed Aprroach

A. System Process Overview

CNN is one om the most famous and commonly used algorithms that has many benefits compared to its predecessors such as it automatically identifies the necessary features without any human supervision [6]. The architecture of CNN is inspired by neurons in the living brain. Thus, this paper uses CNN as the model for its benefit and simplicity. There are steps that the system will go through for the face mask detection system, which can be categorized as input phase, training phase, and output phase. In input phase, the system proceeds to take the input data and prepare the data for preprocessing and split. In training phase, the system will create the CNN architecture for the training. In addition, the system will generate an augmented data that will be flowed to the training set. In output phase, the system then predicts the given input data to predict whether the person is using a mask or not. The system process overview of the face mask detection system is shown in Figure 1.

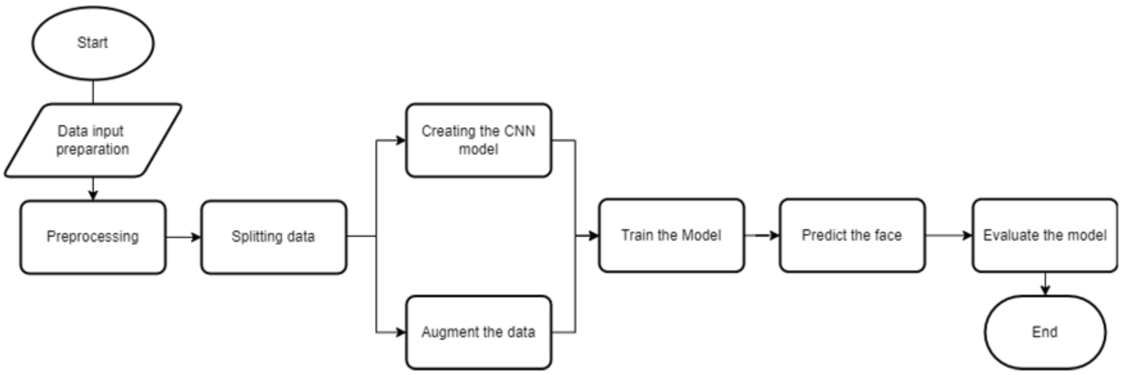

Figure 1. Face Mask Detection system

Figure 1 shows that the approach starts from a given input of an image, then we prepare the data by calling the input data and define the categorical data based on the directory folder of the data path. Then we proceed to preprocess the dataset by implementing the preprocessing methods and normalized the data into 3-Dimensional

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 1, Maret 2023, Pp. 281-290

array consist of (200,200, 3) which consist of 200x200 size of images and 3 color channels. It then proceeds to split the data into training data and target data with a distribution ratio of 70:30. Next is to augment the data to have more diversity and to boost overall performance by using TensorFlow ImageDataGenerator function with 6 parameters which is defined by Zoom Range of 0.4, Rotation Range of 20, Width Shift Range of 0.2, Height Shift Range of 0.2, Horizontal Flip which is equal to True, and Validation Split of 0.2.

After the parameter is set, the augmentation starts, and at the same time it proceeds to create the CNN model with 6 convolutional layers each having a filter of 25, 50, 75, 100, 150, and 200 filter with a kernel size of 3x3. Each filter is convolved during the forward pass across the width and height of the input volume and dot products are computed between the entries of filters and the input position with comprises the final feature map [7]. Then adding a pooling layer that consist of 4 Max Pooling layers with each having a pool size of 2x2, 1 flatten layer and dropout layer with the drop rate of 0.4, and 3 dense layers with each has an output unit of 128, 64 and 2 with the 128- unit dense layer consist of kernel regularizer and bias regularizer that is set to l2(0.01) or L2 regularization factor of 0.01. The final step is to evaluate the model’s performance. Max Pooling is used more widely in CNN architecture which involves taking the maximum value from a region [8]. A max-pooling operator can be applied to down-sample the convolutional output bands, thus reduces variability [9].

B. Low Light Manipulation

The dataset that is used is collected from Prajna Bhandary[1] that consist of 714 normal images of with_mask data and 708 normal images of without_mask data which includes our own dataset that consist of 24 low light images of with_mask data and 22 low light images of without_mask data. We then manipulate the dataset by using Python Imaging Library (PIL) to multiply each pixel by using common multiplication function to gain a low brightness of an image that can be used for the experiment in Section 4. PIL is a library that is originally written by Fredrik Lundh [10] which its successor that is named Pillow which supports Python 3. The result of this manipulation is shown in Figure 2.

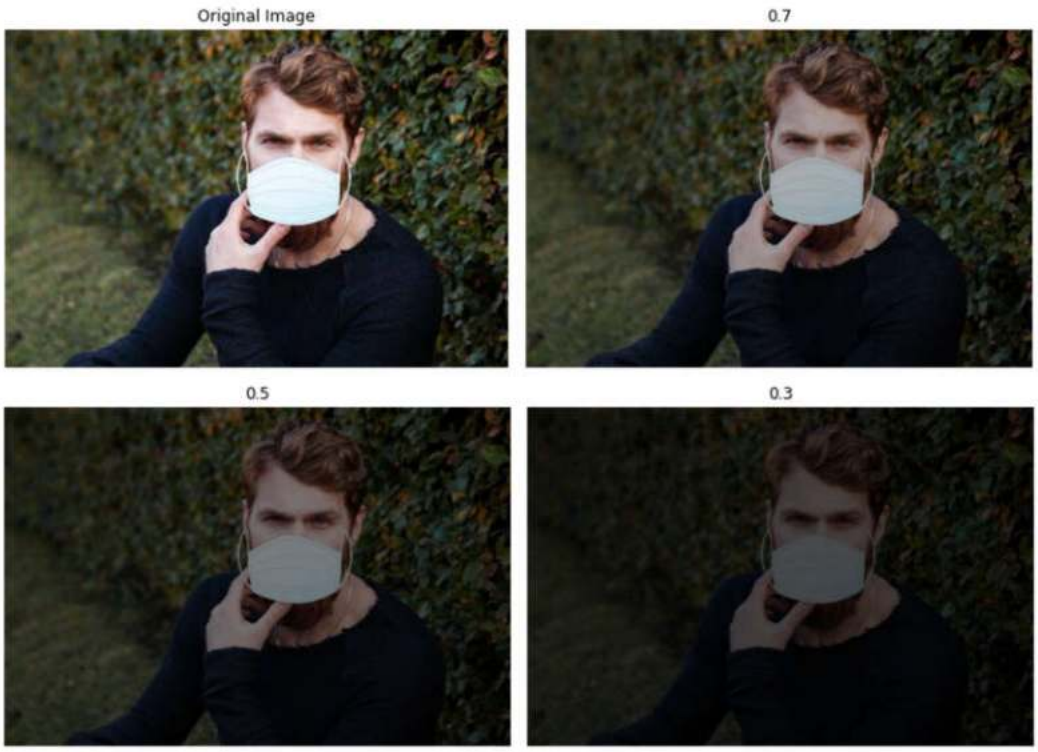

Figure 2. Low light manipulation using level factor of 0.3, 0.5, and 0.7

Based on Figure 2, the manipulation of each pixel can achieve a resemblance of a real low light image. For the simplicity of our research, we limit our research on low light conditions affected only by brightness level. We didn’t consider other factor such as noise and contrast.

[1] https://github.com/prajnasb/face_detector/tree/master/dataset

C. Contrast Limited Adaptive Histogram Equalization (CLAHE)

One of the preprocessing methods that is implemented in this approach is CLAHE. In context, a histogram is a graph that represents the pixel intensity values of an image. Histogram Equalization (HE) is an image processing technique that is used to improved contrast of an image [11]. An 8-bit grayscale images consist of 256-pixel intensities. The color Histogram displays a set of number that represents the pixel distribution among values. It also represents the pixel intensities of a colored image, which is RGB images. The method is implemented by spreading the most of pixel intensity values, which means that it's stretching out the pixel intensity of an image to be equally distributed. This is used to increase the number of low contrast values to gain a higher contrast which can improve the model's performance on training. The main goal for this technique is to give a linear mark to the cumulative probability function (CDF) [12], which we illustrated by computing the low light image with the light level factor of 0.3 using PIL, and we used HE to achieve the result that is shown in Figure 3.

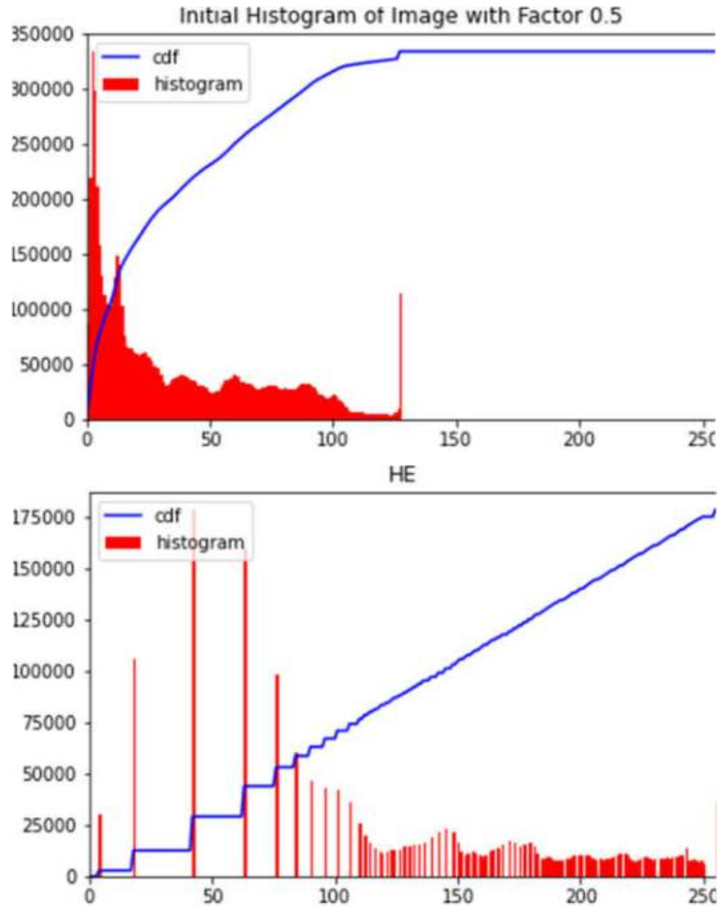

Figure 3. Histogram of the initial image and the equalized image

Figure 3. shows that the initial histogram resides at a darker value, which we can see that the that there are more than 80000 pixels that resides in the intensity value of between 0 and 40. Then we proceed to initialize the Histogram Equalization to the image, and the result for this method is shown such that the darker values are spread/stretch out towards the lighter values by a little, which the implementation result towards an image is shown in Figure 4.

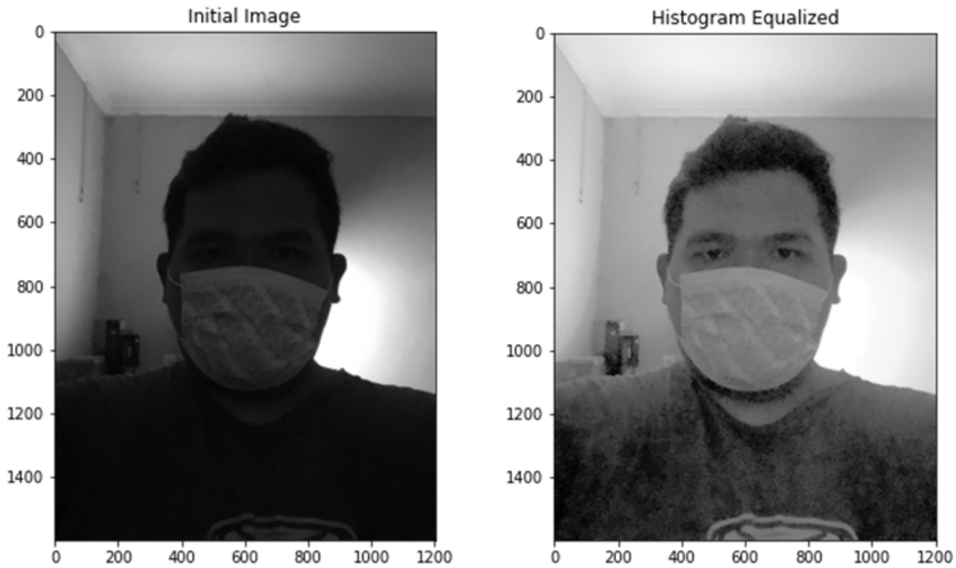

Figure 4. The result of the Histogram Equalization

Adaptive Histogram Equalization (AHE) is different compared to the normal HE in terms of enhancement. AHE enhances the contrast locally by dividing the image into two blocks and computes histogram equalization for each block [13]. AHE can computes multiple histograms with each is corresponding to a section of the image to enhance the local contrast and edges in all regions. The downside of AHE is it tends to increase the number of the contrast in the near-contrast regions which can increase the number of noises in an image. We can limit the increasing number of contrasts by using Contrast Limited Adaptive Histogram Equalization (CLAHE) to reduce the number of noises that is generated from the AHE by clipping the histogram at the predefined value before computing the CDF [14]. The limit for which the histogram is clipped depends on the normalization of the histogram and the size of the neighboring blocks/regions. The result of this method can be shown in Figure 5.

Figure 5. The result of CLAHE

Based on Figure 5, the result of CLAHE is pretty noticeable, which from the look of the eye, we can see that the image gains a sharper identity. This is pretty good results considering without using any noise reduction method and it can be used as an input image for the CNN model to initiate the training phase.

D. Gamma Correction

The image processing methods that will be used to compare the previous method is using Gamma Correction. Gamma Correction is a method that is used to control the overall brightness of an image [15]. Gamma defines the

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 1, Maret 2023, Pp. 281-290

relationship between pixel intensity and the actual luminance. Gamma Correction can be referred to as a power law transform which can be formulated by the following power law expression:

$$O = \left(\frac{I}{255}\right)^{\gamma} x\ 255 \tag{1}$$

where $\gamma < 1$, the original dark areas will be brighter and the histogram will shift to the right. This relation is non-linear; thus, the effect will not be the same for all pixels and it depends on their original value [16]. The result of this power law expression is shown in figure 6.

Figure 6. The results of Gamma Correction using darkened data with low light factor of 0.5 and Gamma (γ) of 3.

Based on Figure 6, using Gamma (γ) of 3 with a darkened data of 0.5 can achieve a brighter image. The result of gamma correction is pretty solid, although we can see that there are noises that is generated within the result image. This is because this image is our dataset that is taken from our mobile phone, and then we manipulate the image to be even darker so we can evaluate how well the CNN model can adapt to this sort of low light image. We can use some sort of noise reduction for this method, but we want to know that if this method can give the model a better performance to learn.

E. Evaluation Metrics

The evaluation metrics that will be used to evaluate the model's performance is by using Confusion Matrix. Confusion Matrix is a matrix that is used to determine the performance of the models by dividing into 2 dimensions which is predicted data and the actual data along with the total predictions that return four different result forecasts [17] which will be shown in Table I.

TABLE I
CONFUSION MATRIX

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 1, Maret 2023, Pp. 281-290

Table 1 shows that the element that is contained within Confusion Matrix that is describe as:

1. True Positive (TP): which the model predicts a positive value and the actual value is positive.
2. True Negative (TN): which the model predicts a negative value and the actual value is negative.
3. False Positive (FP): which the model predicts a positive value and the actual value is negative.
4. False Negative (FN): which the model predicts a negative value and the actual value is positive.

We can use this to evaluate its classification and predictions, and this can inform us on how good our classification model is. Confusion Matrix helps us to calculate different parameters which is describe as:

- Accuracy: used to determine the how often the model predicts positive value.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2}$$

- Precision: used to determine how many were actually positive that the model predicts as positive.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

- Recall: used to determine how many the model predicted correctly from all positive classes.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

- F1-Score: used to measure accuracy of test, and it's calculated from Precision and Recall.

$$F1\ Score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{5}$$

III. Evaluation

TABLE II.
RESULT WITH CLAHE METHOD

Data Type	Metrics			
	Accuracy	Precision	Recall	F1-Score
0.3	97.86%	97.63%	98.09%	97.85%
0.5	97.89%	97.63%	98.09%	97.85%
0.7	98.36%	97.54%	99%	98.26%
Normal	96.49%	97.35%	96.08%	96.71%

We give the model with a batch size of 100 and 100 epochs for training, and we use the input data that is manipulated using PIL with a light level factor of 0.3, 0.5 and 0.7. Then we will also use a normal light image dataset to compare the result of this approach with the same preprocessing method which is CLAHE and Gamma

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 8, No. 1, Maret 2023, Pp. 281-290

Correction. We will evaluate the model's performance using confusion matrix because it's very general in terms of evaluation metrics. The result of the experiment is shown in TABLE II and TABLE III.

TABLE III.
RESULT WITH GAMMA CORRECTION METHOD

Data Type	Metrics			
	Accuracy	**Precision**	**Recall**	**F1-Score**
0.3	98.13%	98.59%	97.68%	98.13%
0.5	98.83%	98.52%	99.01%	98.76%
0.7	98.36%	99.54%	97.30%	98.40%
Normal	97.89%	99.53%	96.42%	97.95%

From the results of the experiment on TABLE IV and TABLE V, we can see that each method produces a different result which is based on the input data. Using CLAHE method with a given set of data can reach a high accuracy of up to 98.36% that is achieved using an input data of 0.7 low light value. Using Gamma Correction method gains an accuracy of up to 98.83% using an input data of 0.5, which is slightly higher than using CLAHE method that gains an accuracy of 97.89%.

Then we proceed to compare the propose model with a set of pre-trained models namely MobilenetV2, VGG16 and VGG19. These pre-trained models are trained with both 100 epochs and will be given a dataset of image with light level factor of 0.3. We proceed to used CLAHE and Gamma Correction as a comparison for the preprocessing methods, and the result of the experiment can be shown in TABLE IV and TABLE V.

TABLE IV.
EVALUATION WITH CLAHE METHOD

Model	Metrics			
	Accuracy	**Precision**	**Recall**	**F1-Score**
MobileNetV2	99.06%	98.60%	99.53%	99.06%
VGG16	98.59%	98.24%	99.11%	98.67%
VGG19	98.36%	98.16%	98.61%	98.38%
Proposed	**97.86%**	**97.63%**	**98.09%**	**97.85%**

TABLE V.
EVALUATION WITH GAMMA CORRECTION METHOD

Model	Metrics			
	Accuracy	**Precision**	**Recall**	**F1-Score**
MobileNetV2	98.36%	98.59%	98.20%	98.39%

VGG16	96.96%	96.29%	97.65%	96.96%
VGG19	94.61%	90.95%	97.94%	94.31%
Proposed	**98.13%**	**98.59%**	**97.68%**	**98.13%**

Based on the experiment result of TABLE IV and TABLE V, the result of the proposed model can achieve an accuracy that is almost comparable towards the other pre-trained models. Using CLAHE method, the accuracy values of the proposed model with the other pre-trained model is slightly lower with the range difference of 0.5% with VGG19, 1.2% with MobileNetV2, and 0.73% with VGG16. Using Gamma Correction method, the range difference of the accuracy values of the proposed model with the other pre-trained model is 0.23% lower with MobileNetV2, while VGG16 and VGG19 gains lower accuracy than the proposed method with the range difference of 1.17% with VGG16, and 3.52% with VGG19.

From the result analysis that is conducted, it can be compared with the previous work result that is related to the Face Mask Detection which uses the same CNN method with a simplified approach of CNN architecture using normal input image. The method gains an accuracy of 95.77% and 94.58% respectively using two different datasets. Other previous work that implements the Face Mas Detection using ResNet50 that uses MAFA (MAsked FAces) dataset which gains an accuracy of 98.2%. Another previous work that uses Single Shot Multibox along with MobileNetV2 as its architecture which gained an accuracy of 0.9264 and an F1-Score of 0.93.

IV. CONCLUSION

We have conducted research to build a face mask detection system under low light condition. We used CNN model by using PIL to manipulate the pixel within the image to create a resemblance of a low light image. The proposed approach achieved an accuracy of 97.66% and 98.13% using CLAHE and Gamma Correction as a pre-processing method which state that this approach can be viable to be used for a low light image detection. The model then compared with a pre-trained model such as MobileNetV2, VGG16 and VGG19 which shows good results in terms of accuracy, precision, recall, and f1-score.

Further Improvements of the experiment will be conducted to evaluate the model's performance in this case and finding more alternative methods which can help with cases such as low light to improve performance and classification of the model.

REFERENCES

[1] P. Nagrath, R. Jain, A. Madan, R. Arora, P. Kataria and J. Hemanth, "SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2," Sustain Cities Soc., 2020.

[2] A. Das, M. W. Ansari and R. Basak, "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV," IEEE, India, 2020.

[3] S. Sethi, M. Kathuria and T. Kaushik, "Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread," J Biomed Inform., 2021.

[4] J. Yu, X. Hao and P. He, "Single-stage Face Detection under Extremely Low-light Conditions," in *IEEE*, Montreal, BC, Canada, 2021.

[5] W. Chen and T. Shah, "Exploring Low-light Object Detection Techniques," *arXiv,* vol. 1, no. Computer Vision and Pattern Recognition (cs.CV), p. 5, 2021.

[6] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Samantamaria, M. A. Fadhel, M. Al-Amidie and L. Farhan, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data,* vol. 8, no. 1, 2021.

[7] T. Gorach, "DEEP CONVOLUTIONAL NEURAL NETWORKS- A REVIEW," *International Research Journal of Engineering and Technology (IRJET),* vol. 05, no. 07, pp. 439-452, 2018.

[8] N. A. Samat, M. N. B. Mohd Salleh and H. Ali, "The Comparison of Pooling Functions in Convolutional Neural Network for Sentiment Analysis Task," in *Recent Advances on Soft Computing and Data Mining*, Springer, 2020, pp. 202-210.

[9] H. Gholamalinejad and H. Khosravi, Pooling Methods in Deep Neural Networks, a Review, 2020.

[10] P. Podrzaj and S. Simoncic, "IMAGE PROCESSING CAPABILITIES OF PYTHON," *International Journal of Mechanical and Production Engineering,* vol. 6, no. 10, pp. 77-81, 2018.

[11] S. Sudhakar, "Histogram Equalization," Towards Data Science, 10 July 2017. [Online]. Available: https://towardsdatascience.com/histogram-equalization-5d1013626e64#:~:text=Histogram%20Equalization%20is%20a%20computer,intensity%20range%20of%20the%20image.. [Accessed 1 August 01].

[12] M. G. W. A.-S. I. S. Irem Doken, *Histogram Equalization Of The Image,* arXiv, 2021.

[13] K. S. Htoon, "A Tutorial to Histogram Equalization," medium, 19 August 2020. [Online]. Available: https://medium.com/@kyawsawhtoon/a-tutorial-to-histogram-equalization-497600f270e2. [Accessed 2 August 2022].

[14] Pintusaini, "Adaptive Histogram Equalization in Image Processing Using MATLAB," *MATLAB image-processing,* p. 1, 22 November 2021.

[15] R. Sachdeva, Sonam and H. Sharma, "Face Mask Detection System," International Journal of Scientific and Engineering Research, 2020.

[16] F. Amer and M. S. H. Al-Tamimi, "Face Mask Detection Methods and Techniques: A Review," *ResearchGate,* pp. 3812-3825, 2022.

[17] J. D. Novakovic, A. Veljovic, S. S. Illic, Z. Papic and M. Tomovic, "Evaluation of Classification Models in Machine Learning," *Theory and Applications of Mathematics & Computer Science 7,* vol. 1, pp. 39-46, 2017.