Enhanced Real-Time Detection of Face Mask with Alarm System using MobileNetV2

Christian Lloyd Amar*1, John Ronelo Menasalvas2, Aaron Jake Candido3, Cuburt Balanon4, Hajah Sueño5

Computer Studies Department, Notre Dame of Marbel University, Koronadal City, Philippines
amar.clloyd900@gmail.com1, johnronelo17@gmail.com2, ajcandido09@gmail.com3, balanoncuburt@gmail.com4, htsueno@ndmu.edu.ph5

*Corresponding Author:-
amar.clloyd900@gmail.com

Abstract—To prevent the coronavirus from spreading, the government adopted measures such as wearing a face mask in public locations. The researchers aimed to create a face detection system using the MobilenetV2 architecture that would identify a person’s faces and determine whether they were wearing a face mask. The built model will help to reduce the danger of viral transmission. In this study, face mask detection is achieved using a machine learning algorithm and the classification method using MobileNetV2. The steps for building the model are data gathering, data pre-processing, splitting the data, testing the model, and implementing the model. The built model can distinguish between those who are wearing a face mask (with no design patterns) and those who are not wearing it with a 96% accuracy. In terms of classification accuracy, the proposed model using MobileNetV2 outperformed the other models LeNet-5, AlexNet, and ResNet-50. If the detected person is labeled with “no mask”, the system generates an alarm sound. This research will be useful in combating virus spread and avoiding virus contact.

Keywords—image processing, mobilenetv2, face detection, face mask
I. INTRODUCTION

COVID-19 was declared a global pandemic by the World Health Organization on March 11, 2020 [1]. The global number of COVID-19 cases was increasing daily at the time of the announcement, and it was still increasing five months later. Many countries, including the Philippines, took a breather, went into lockdown, or hunkered down. As a result, several towns were told to stay home except to get vital supplies.[1]. The government adopted rules such as wearing a face mask in public places to prevent the coronavirus from spreading in the fight against this deadly virus Masks, in particular, have been shown to be helpful in both preventing sickness and decreasing asymptomatic transmission in healthy people [2]. It became necessary to establish a protocol requiring people to wear masks before accessing public areas [3].

Some researchers created a face mask detecting system that can tell whether someone is wearing one. However, there are no effective face mask detection devices available to assess whether someone is wearing one[4]. Thus, utilizing Convolution Neural Network (CNN), which is commonly used in image classification and identification, this study created a real-time face detection system that can accurately detect the face under the mask in transits. The system will help to reduce viral transmission risk.

II. LITERATURE REVIEW

A. Transmission of COVID-19 in Public Places

The global pandemic of Coronavirus Disease 2019 (COVID-19) spread rapidly, posing a major threat to global public health. According to the World Health Organization, COVID-19 has infected over 200 countries and territories, with over 19.46 million confirmed cases by August 9th, 2020. (WHO) [5]. Because of the pandemic’s global spread, long-term effective prevention and control techniques for varied settings and vulnerable groups should be adopted during this pandemic. Public transit is an important form of transportation for many individuals, and in other situations, it is their only mode of mobility. Public transportation vehicles have restricted spaces that are suitable for the transmission of infectious diseases from person to person. As a result, multiple countries have reported numerous clusters of infections in infectious respiratory viruses, including SARS Coronavirus 2, which has been identified in public transportation. People who do not develop flu symptoms within five days of symptom onset are still contagious, increasing the risk of influenza infection. [6].

COVID-19 has been linked to bus travel in several research. [7]. Following an increase in the number of reported incidents, one country after another adopted so-called “social distancing” policy that affected schools, stores, workplaces, public transit, and a variety of other locations [8]. Early estimates suggest that during the COVID-19 pandemic, passengers dropped by as much as 80%–90% in important cities in China, Iran, and the United States, and as much as 70% for some operators in the United Kingdom. [4].

While some of the initial concerns about using public transit have dissipated, riders remain wary, and there is more concern about public transportation hygiene than there was before [9]. Risk perceptions may thus influence not only current travel decisions and time-versus-crowding trade-offs, but also long-term travel decisions [10]. As a result, anticipating demand during the various stages of this extraordinary crisis is fraught with uncertainty.

B. Face Mask Detection

The face mask detection model has become one of the most necessary models during the COVID-19 pandemic. From the manual monitoring of whether people in public and crowded areas wear a face mask, [11] an automated model has been developed in real-time in a research project that is integrated into surveillance cameras in the public and detects if people in public areas wear face masks and maintain social distance and inform the relevant authorities through the use of computer vision. The coronavirus has caused worldwide health crises and the wearing of masks became the basic prevention against the virus [5].

Several studies have been conducted which focused on the detection of human faces where they are wearing masks [12] [13] [14] [15]. They discovered that when a face mask is worn, the accuracy of human face detection drops by 70%, and they created a technique to determine how someone is wearing the face mask. They were able to classify three types of facemask-wearing situations: accurate facemask-wearing, incorrect facemask-wearing, and no face mask-wearing. In [16] the authors used a GAN-based network with two discriminators to remove the mask from a face and reconstruct the face without a mask using the CelebA dataset.

C. Facial Features Detection

Face detection is the first step in any automated system that attempts to solve problems like face identification, face tracking, or facial expression recognition. [17]. In the study of [18], they looked at objects in the same row and classified them as eyeballs by calculating the half-distance between them and determining whether there is another object at this location, which may be the nose or mouth if these features are present. In a random collection of samples, the human face, eyes, nose, and mouth are recognized and tested. Examining the distance between the eyes and matching the pupil helps identify the human’s left and right eye pairs, the nose with the darker region on two sides and the brighter section in the center, the mouth, and the face with numerous points on it. The purpose of facial key point recognition is to learn where the brows, eyes, mouth, and nose are on the face. When an image is given as an input, the algorithm finds the distance between the two eyes first, then processes the algorithm to match the eyes distance and pupil distance, resulting in the detection of the eyes. The serial eyes, nose, mouth, and face are detected using the same matching method [19].

D. Face Mask Detection Algorithm

Face mask wear can be detected using two key elements. The first step is to figure out where the face is in the image; the second is to figure out if the face in the data set is wearing a mask and if it’s worn correctly. Face occlusion, variable face scale, uneven illumination, density, and other issues affect the existing object detection approach, and these issues have a substantial impact on the algorithm’s performance. Furthermore, the usual object detection algorithm uses
selective search approach for feature extraction, which leads to difficulties including poor generalization capacity, redundant information, low accuracy, and poor real-time performance[19]. To recognize face masks, some researchers have used RGB color information extraction [20]. However, because the study does not address the issue of using non-standard masks, the algorithm's adaptability must be increased. The study of [21] achieved face mask recognition by combining YOLO-v2 and ResNet50, with DarkNet-19 as the backbone network. DarkNet-19, on the other hand, has been optimized by CSPDarkNet53.

The ablation experiment revealed that the CSP1 X module outperforms the CSPDarkNet53 module in our research. Face masks may be recognized with 99.64 percent accuracy when ResNet50 and SVM are combined. However, the method entails high computing costs[5]. Furthermore, a paper [22] proposed combining SSD with MobileNetV2 for mask recognition, however, its model structure is too complex and its performance is inferior to YOLO-v4. Two categories are employed, and the impact of wearing masks irregularly on the algorithm was not evaluated. As a result, these algorithms' feature extraction capability and model practicability must be improved.

Face mask identification is evaluated in this study, which uses enhanced YOLO-v4 and includes three categories: face mask, face, and WMI. Furthermore, CSP1 X enhances the ability of this article to extract features, but CSP2 X encourages PANet to speed up the circulation of semantic features and increase feature fusion, hence increasing the model's robustness.

E. **MobileNetV2**

With the advancement of technology, developing trends, new trends, and new methodologies. Several researchers used technology to contribute to public health and wellbeing. Using the MobileNetV2 architecture, researchers were able to construct a more efficient and accurate face mask detection model that can be used in a variety of high and low computational applications. The trained model was evaluated on both real-time videos and static images, and it performed better in both scenarios than the other planned model. The suggested model on face mask detection was completed using the model constructed with CNN architecture utilizing MobileNetV2, which yielded an excellent result with flawless detection accuracy [23].

F. **The Model Structure of the YOLO-v4 Network**

YOLO-v4 is a high-precision and real-time one-stage object detection algorithm based on regression proposed in 2020, which combined the characteristics of YOLO-v1, YOLO-v2, YOLO-v3, and others to attain the current optimum in terms of detection speed and detection accuracy trade-off. Figure 1 depicts the model structure, which is divided into three parts: the Backbone, the Neck, and the Prediction [24].

When combined with the ResNet structure's properties, YOLO-v3 absorbed the residual module into itself, yielding Darknet53. Taking into account the higher learning ability of Cross-Stage Partial Network (CSPNet) [25] YOLO-v4 built the CSPDarkNet53. The feature layer is sent into the residual module, and the higher-level feature information is output.

![YOLO-v4 Network Structure](image)

The difference between the output and the input becomes the model's learning goal in the ResNet module, resulting in residual learning while reducing model parameters and strengthening feature learning. The Neck is made up of the SPPNet and the PANet. The input feature layer in SPPNet is convolved three times before being maximally pooled utilizing the greatest pooling cores available. The pooled findings are first concatenated, then convolved three times, resulting in an improved network receptive field. PANet concatenates the feature layers obtained by Backbone and SPPNet with the feature layers obtained by CSPDarkNet53 to realize feature fusion, and then down-samples, compressing the height and width, and finally stacking with the previous feature layers to realize more feature fusion (five times).

The prediction module can create predictions based on the network properties retrieved. Using a 13 x 13 grid as an example, partitioning the input image into 13 x 13 grids, and then pre-setting each grid with three prior frames is equivalent. The network's prediction outputs will be used to change the placements of the three prior frames before being filtered using the non-maximum suppression (NMS) [26] algorithm to generate the final prediction frame.

III. **METHODOLOGY**

In this study, the researchers enhanced the Convolutional Neural Network (CNN) model in detecting whether the person is wearing a face mask or not by utilizing MobileNetV2 for fine-tuning. MobileNetV2 is chosen because it has various benefits, including the following: 1) it is a lightweight Deep Learning appropriate to edge devices, 2) it gives
outstanding results for object identification, and 3) it can easily tradeoff between accuracy and latency using simple global hyperparameters [27]. As shown in Figure 2, the procedures to implement and validate the performance of MobileNetV2 include data gathering and preprocessing, splitting the data, building the model, and evaluation.

A. Data Gathering

The dataset is being used to train data on people who wear plain face masks and those who don’t. This study used 453 data with masks and 353 data without masks to develop the model. The image has been cropped so that only the object's face is visible. The obtained data were classified into two groups: with and without a mask.

![Fig. 2. Enhanced CNN Model using MobileNetV2 Architecture](image)

B. Preprocessing

The preprocessing phase consists of four steps: resizing the image, converting the image to an array, preprocessing the input using MobileNetV2, and performing hot encoding on labels. Due to the obvious effectiveness of training models, image resizing is a critical pre-processing step in computer vision. If the image is smaller, the model will perform better. In this experiment, enlarging the image resulted in a 224 by 224-pixel image. All the photos in the dataset are converted into an array using these techniques. The image is transformed into an array so that it may be called by the loop function. The image will then be utilized to use MobileNetV2 to pre-process input. Because many machine learning algorithms cannot function on data labeling instantly, the final step in this phase is to execute hot encoding on labels. It requires that all input and output variables be numeric, including this process. The tagged data will be turned into a numerical label, which the algorithm will be able to understand and interpret.

C. Splitting the Data

Following the pre-processing step, the data is divided into two batches: training data (80 percent) and testing data (the remaining 20 percent). Each batch includes both with-mask and without-mask photos.

D. Building the Model

The model is constructed in the following steps: the first is data augmentation, which involves using the image generator to enhance the image. The image is reversed horizontally with the rotation range set to 20, the zoom range set to 0.15, the width and height shift range set to 0.2, the sheer range set to 0.15, and the sheer range set to 0.15. The following step involves loading pre-trained ImageNet weights into MobileNetV2. To protect previously learned attributes, the base layers are then frozen. Following that, it adds new trainable layers to the model, which are trained on the obtained dataset to further increase the model's classification accuracy in determining whether a person is wearing a mask or not. After that, the model is fine-tuned, and the weights are saved. Using a pre-trained model improves processing time and allows to use of existing biased weights without having to delete previously learned features. The model can now be compiled using the Adam optimizer, which is recommended because it uses less memory and has a faster convergence rate. The model's head is trained for 20 epochs with a batch size of 32. The goal of this stage is to find a set of weights and biases that cause the least amount of average loss across all scenarios. Finally, the model is kept so that it can be utilized for future prediction and implementation.

E. Real-time Detection of the Mask Mask using the Trained Model

A live video stream through the camera is used at this stage. To capture live video, OpenCV is imported. OpenCV will then collect real-time data from the camera. The system can identify 5-10 faces simultaneously. The real-time data (frames per second) is obtained from the camera to categorize it and predict the output of the given real-time input using the trained model. As a result, we obtain a frame in which if a person is wearing a mask, it shows "Mask" with a green rectangle on his/her face and if a person is not wearing a mask, it shows "No Mask" with a red rectangle on his/her face. If a certain individual is not wearing a mask, the system automatically activates the alert system to avoid the irresponsibility of not wearing a mask.
**F. Evaluation Metrics**

To assess the proposed model's performance, classification techniques and performance measures are necessary. The number of correct predictions divided by the total number of predictions generated for a dataset is known as Accuracy. Precision is an acceptable measure of determination, but the cost of false positives is high, and the F1 score is required when you want symmetry between precision and recall, and Recall is the model metric used to select the best model when there is an elevated cost associated with false-negative predictions [5].

\[
\text{Accuracy} = \frac{TP + TN}{(TP + FP) + (TN + FN)}
\]  
\[
\text{Precision} = \frac{TP}{TP + FP}
\]  
\[
\text{Recall} = \frac{TP}{TP + FN}
\]  
\[
\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

Where TP = True Positive, TN = True Negative, FP = False Positive, FN = False Negative

**IV. RESULTS AND DISCUSSIONS**

The result of this experiment was conducted using a laptop equipped with Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80GHz, 12GB RAM, and 64-bit Windows 10 OS. The researchers also used Jupyter Notebook equipped with Python 3(ipykernel) and utilized needed libraries for the development and model evaluation. The datasets are collected from which are labeled “with mask” and “without a mask” separately. For quicker calculation, the images are pre-processed and converted to NumPy arrays. The first dataset was divided into two parts: 80% training and 20% testing. The second dataset continues the dataset's partition, which includes the training set, validation set, and testing set. To generate variances in the data and avoid overfitting, data augmentation was utilized and used on the training sets. To ensure that the model can predict well, it is tested by making predictions on the testing set.

Figure 3 shows the prediction result of testing the model.

Table 1 shows the results of 20 iterations of evaluating the loss and accuracy after training the model.

<table>
<thead>
<tr>
<th>Epochs</th>
<th>Loss</th>
<th>Accuracy</th>
<th>Val Loss</th>
<th>Val Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/20</td>
<td>0.5727</td>
<td>0.7585</td>
<td>0.3756</td>
<td>0.8707</td>
</tr>
<tr>
<td>2/20</td>
<td>0.3661</td>
<td>0.8703</td>
<td>0.2589</td>
<td>0.9211</td>
</tr>
<tr>
<td>3/20</td>
<td>0.2742</td>
<td>0.9117</td>
<td>0.2114</td>
<td>0.9430</td>
</tr>
<tr>
<td>4/20</td>
<td>0.2234</td>
<td>0.9303</td>
<td>0.1805</td>
<td>0.9464</td>
</tr>
<tr>
<td>5/20</td>
<td>0.2012</td>
<td>0.9457</td>
<td>0.1607</td>
<td>0.9464</td>
</tr>
<tr>
<td>6/20</td>
<td>0.1781</td>
<td>0.9489</td>
<td>0.1462</td>
<td>0.9495</td>
</tr>
<tr>
<td>7/20</td>
<td>0.1617</td>
<td>0.9530</td>
<td>0.1360</td>
<td>0.9464</td>
</tr>
<tr>
<td>8/20</td>
<td>0.1488</td>
<td>0.9554</td>
<td>0.1275</td>
<td>0.9495</td>
</tr>
</tbody>
</table>

Fig 3. Confusion Matrix
Table 2 shows the model's evaluation, which shows the model's precision, recall, F1 score, and accuracy.

**TABLE II. Model Evaluation**

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>With mask</td>
<td>0.95</td>
<td>0.99</td>
<td>0.97</td>
<td>238</td>
</tr>
<tr>
<td>Without mask</td>
<td>0.97</td>
<td>0.85</td>
<td>0.91</td>
<td>79</td>
</tr>
<tr>
<td>Accuracy</td>
<td></td>
<td></td>
<td>0.96</td>
<td>317</td>
</tr>
<tr>
<td>Macro avg</td>
<td>0.96</td>
<td>0.92</td>
<td>0.94</td>
<td>317</td>
</tr>
<tr>
<td>Weighted avg</td>
<td>0.96</td>
<td>0.96</td>
<td>0.95</td>
<td>317</td>
</tr>
</tbody>
</table>

Figure 4 shows the model's implementation, which uses the trained model to predict the input data. Figure 5 shows a real-time test of the model’s output in detecting multiple faces simultaneously.

**Fig. 4. Model Implementation**

**Fig. 5. Simultaneous Detection of Multiple Faces**

A. **MobileNetV2 Experimental Results**

The results of testing the model’s loss and accuracy after 20 iterations for epochs with a batch size of 32, as shown in Table 1. After testing the model, a confusion matrix is built with values shown in Figure 3. The average accuracy came out to be 96% for predicting if a person is wearing a face mask or not on a dataset, as shown in Table 2. Figure 5 shows the different test results for the model's performance in detecting persons labeled "mask" and "no mask". The rectangle green box indicates that the targeted face is correctly wearing a face mask, whereas the rectangular red box
indicates that the targeted face is not properly wearing a face mask or is not wearing a face mask with the accuracy indicated on top.

B. Proposed Model Comparison with Existing Face Mask Classifiers

As shown in Table 3, the proposed model’s performance is compared to other recent studies. LeNet-5, AlexNet, and ResNet-50 were selected for this comparison. The result concludes that the proposed model outperformed the other models in terms of classification accuracy which is 96%.

<table>
<thead>
<tr>
<th>Architecture Used</th>
<th>Accuracy</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet-5[28]</td>
<td>84.6%</td>
<td>0.85</td>
</tr>
<tr>
<td>AlexNet[29]</td>
<td>89.2%</td>
<td>0.88</td>
</tr>
<tr>
<td>ResNet-50[30]</td>
<td>92.7%</td>
<td>0.92</td>
</tr>
<tr>
<td>MobileNetV2(Proposed Model)</td>
<td>96.0%</td>
<td>0.96</td>
</tr>
</tbody>
</table>

V. CONCLUSION

Based on the results of the study, the following conclusions are produced: CNN face detection’s capability has been enhanced in a way that MobileNetV2 was utilized, and the system triggers an alarm if the person is not wearing a face mask. The proposed model is tested in a real-time environment which satisfies the following parameter: Detected up to 5 faces simultaneously and within 4 meters range, accuracy is maintained at 96% percent.

The classification performance of MobileNetV2 in detecting face masks is calculated, and the following results are obtained the average accuracy and precision achieved 0.96 while the recall has achieved 0.92 and 0.94 for the F1-score. The proposed model was compared to other CNN mobile architectures and the proposed model (MobileNetV2) outperformed the LeNet-5, AlexNet, and ResNet-50, with an average classification accuracy of 96%. Based on the preceding conclusions, it is shown that the requirements have been achieved, and with the usage of MobileNetV2 architecture, the researchers enhanced the face mask detection system for detecting faces in transit and detecting whether the person is wearing a mask or not in a real-time.

ACKNOWLEDGMENT

We would like to express our gratitude to the Notre Dame of Marbel University for the generosity and support in the completion of this research.

REFERENCES


