EasyChair Preprint

№ 9336

Investigation and Analysis of Deep Learning and Machine Learning Algorithm for Face Mask Detection System

Muhammad Ezzuddeen Suratman and Fahmi Samsuri

November 18, 2022

Applied Mathematics and Computational Intelligence
Volume X, Jan 2022 [xx – xx]

Investigation And Analysis Of Deep Learning And Machine Learning Algorithm For Face Mask Detection System

Muhammad Ezzuddeen bin Suratman[1], Fahmi bin Samsuri[2]

[1,2]College of Engineering, Universiti Malaysia Pahang, 26300 Gambang, Pahang, Malaysia

[1]mezzuddeen1997@gmail.com, [2]fahmi@ump.edu.my

Received xx January 2022; Received in revised xx July 2022; Accepted xx September 2022
Available online xx December 2022

ABSTRACT

People nowadays tend to wear a protective facemask because of the pandemic COVID-19 that strike our world few years ago and wear protective facemask has become a new normal. Many public place that provides a certain service want people to wear mask correctly before entering the place. Therefore, by developing the facemask detection system, it tends to help a global society to aware the environment that surround by the virus and to prevent the infections. Although vaccines have been developed, people still need to be aware because of some society that stick not to wanting a vaccine. For develop this system, machine learning and deep learning is the best method to use by using some basic machine learning package such as Tensorflow, Keras and OpenCV. This method detects the image of someone face from the image, video and real time monitoring correctly and then identifies it has a facemask on it or not and will alert the authority if not wearing a facemask. This system can be use at the premise before people entering the place and would eliminate the need to place a worker to monitor the people coming in at the entrance and minimize the infections.

Keywords: Object Detection, real time, machine learning, deep learning, OpenCV.

1 INTRODUCTION

For the past 2 years, the world has been strike by the new type of virus that called COVID-19 and new norm have been implemented. Because of that, people have been forced to wear a face mask and keep sanitize to prevent the virus spread. The covid-19 virus has brought about a new normal existence in which social distance and the use of face masks play an important part in restricting the virus's spread. However, most people do not wear face masks in public place, which contributes to the transmission of infections. To avoid such situations, we must sanitize and make people aware about the importance of wearing face masks. Humans are not permitted to participate in this process due to the risk of being affected by virus. Face mask detection will be determining whether or not someone is wearing a mask, comparable to detecting any object in a scene and then determining whether or not it has a mask on it. Many methods for object detection have been proposed, and deep learning give the best result when come to detect the object.

Face mask detection entails identifying the position of the face and then deciding whether or not it is wearing a mask and wearing mask incorrect. The problem is similar to broad object detection in terms of detecting object types. Face identification is concerned with differentiating a certain collection of things, namely faces. It has a wide range of uses, including autonomous driving, teaching, and surveillance[1]. This paper presents a simplified approach to serve the above purpose using the basic Machine Learning (ML) packages such as TensorFlow, Keras, and OpenCV by showing each of model used in machine learning for best accuracy result.

2 RELATED WORK

Because of the enormous threat posed by the COVID-19 pandemic, creative strategies to optimise the inappropriate use of preventative systems have become a hot issue. AI is one of the most helpful strategies for adapting tasks like picture categorization to changing circumstances. In this section, we will discuss the most important studies on AI and COVID-19 technological solutions. By recognising patterns in photos, AI and CNN solutions may address issues. Author in [2] created an application integrated with a mobile phone capable of detecting and categorising plants through the use of photographs using InceptionV3 CNN. They have also developed a prototype for recognising tree species that performs real-time classification remotely.

The authors of [3] created a mobile application that determine the service life of a facemask, showing what phase it is in and how effective it is after a period of use. They achieve this by collecting four grey features from microphotographs of the facial mask and using co-occurrence matrices (GLCM) to do so. Three outcomes are obtained using KNN. These have an accuracy of 82.87 percent. The accuracy of "normal usage" and "not advised" is 92.00 and 92.59 percent, respectively.

According to the research published in [4], convolutional neural networks (CNNs) in computer vision have a strong limit on the size of the input picture. To overcome the inhibition, the common approach is to reorganise the pictures before putting them into the network

3 MATERIAL AND METHODS

3.1 Dependencies

The dependencies (packages) or a framework must be installed. TensorFlow, Keras, which covers TensorFlow's numerical calculation libraries and allows you to design and train only a few lines of code, and OpenCV are the required dependencies.

A. TensorFlow

TensorFlow, an interface for expressing machine learning algorithms, is used to implement ML systems into fabrication in a variety of computer science fields, including sentiment analysis, voice recognition, geographic information extraction, computer vision, text summarization, information retrieval, computational drug discovery, and flaw detection. TensorFlow is used as the backend for the entire Sequential CNN architecture (which consists of several layers) in the proposed model. It is also used in data processing to reshape the data (image).

B. Keras

Keras provides fundamental reflections and building units for the creation and delivery of ML arrangements at a high iteration velocity. It fully utilises TensorFlow's scalability and cross-platform capabilities. Keras' primary data structures are layers and models. Keras is used to implement all of the layers in the CNN model. It aids in the compilation of the overall model in conjunction with the conversion of the class vector to the binary class matrix in data processing.

C. OpenCV

OpenCV (Open Source Computer Vision Library), an open-source computer vision and ML software library, is used to differentiate and recognise faces, recognise objects, group movements in recordings, trace progressive modules, follow eye gesture, track camera actions, expel red eyes from pictures taken with flash, find comparative pictures from an image database, perceive landscape and set up markers to overlay it with enhanced reality, and so on. The proposed method makes use of these OpenCV features in the resizing and colour conversion of data images.

3.2 Datasets

Dataset from the GitHub consist of 6000 images and has been classify into three different classes which is 2000 image of people wearing facemask, 2000 without facemask and 2000 images with mask incorrect. All the image has many variations of size and resolution. Because of that the image will be resize into 64x64 in data pre-processing.

Figure 1 : Without face mask dataset

Figure 2 : Wearing face mask dataset

Figure 3 : Wearing face mask incorrect dataset

3.3 Data Pre-Processing

Pre-processing is the stage in which the image is resized. The image would be enormous once captured hence it would be easy. Transforming the data and utilising it into another process is called data pre-processing. These structured data may be used with an information model or composition and record the relationships between multiple entities. Pre-processing is the part where the resizing of the image is involved. The image would be large once captured therefore it would be easy. Data pre-processing is the process of converting the data and used it into another process. These organised data are compatible with model information[5]. If the images were smaller than 80x80 pixels, and the faces were even smaller, the image is scaled up by a factor of 2 or more. The image is resized into (224x224) pixels to get the best result.

A. Visualised the data

Process to translate abstract data into meaningful representations for knowledge exchange and insight finding. It is beneficial to investigate a certain trend in the dataset[6]. The total number of images in the dataset is visualised in 'with mask', 'with mask incorrect' and 'without mask' categories. The os.listdir(path) statement categorises the list of directories in the specified data path. Then it will proceed with convert image to grayscale and resizing the image. In this project, image size have been set to 64x64.

B. Conversion RGB to Grayscale

Modern descriptor-based image recognition systems routinely work on grayscale images without delving into the method used to convert from RGB image to grayscale. This is due to the fact that when using robust descriptors, the color-to-grayscale technique has little effect. Non-essential data may increase the quantity of training data required for exceptional performance. Grayscale is used to extract descriptors rather than working on colour photographs straight immediately since it streamlines the process and reduces computer requirements[7].

C. Reshaping Image

During image relegation, the input is a three-dimensional tensor with a prominent unique pixel in each channel. All the images must be the same size and correspond to the same 3D feature tensor[8]. However, neither images nor their corresponding feature tensors are normally coextensive. Most CNNs can only accept images that have been fine-tuned. This causes a number of issues during data collection and model implementation. However, reconfiguring the input images before augmenting them in the network can assist in overcoming this limitation.

3.4 Train Model

In this stage, certain parameter will be configured in KNN, SVM, Decision Tree and CNN model to determine the accuracy best model suit for the system.

A. K-Nearest Neighbour (KNN)

The number of neighbours and the kind of distance are the parameter settings utilised in the KNN method. The number of neighbours must be an odd number utilised as 5, 7 and 9 and the distance types are Minkowski. Minkowski will measure designed for vector spaces with real values. Normed vector space will give the result calculate for distance which refer to the space where distances length must be positive.

$$\left(\sum_{i=1}^{n}|x_i - y_i|^p\right)^{\frac{1}{p}} \tag{1}$$

Equation (1) show the general form for Minkowski distance and different distance can be get by manipulated the value of p. When set the value of p = 1 will give Manhattan distance and set p = 2 will give Euclidean distance. Because using Python library SKlearn so the default setting is 5 neighbour and Euclidean distance.

B. Support Vector Machine (SVM)

Optimum hyperplane can be determined by increase the distance between the classes. The maximum distance between data items in each class is defined as the distance between classes[9]. Figure 4 shown the illustration to define hyperplane.

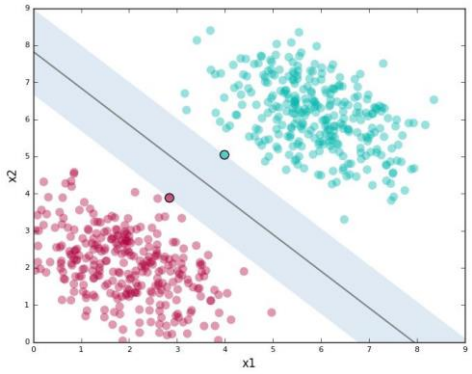

Figure 4 : Classification to define hyperplane using SVM

When compared to KNN, SVM performs better with large dimensionality features by producing a hyperplane to split data with a straight line. Red and Blue dot indicate the classes of the data then SVM will strive to determine the optimal dividing line boundary between the two sets of data and this is the process of learning for SVM[9].

C. Decision Tree

For this model, decision tree is used to calculate the entropy and information that have been gain. Computing the entropy yields the amount of uncertainty in the data, whereas calculating the difference in the entropy yields the information gain.

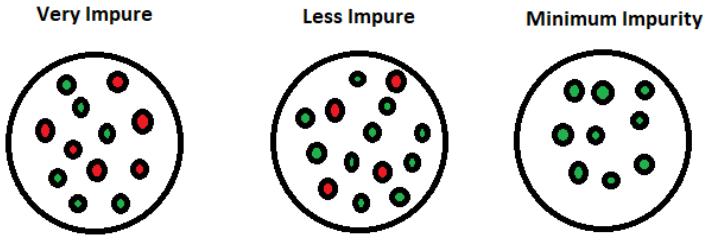

Figure 5 : Impurity of data in Decision Tree

Entropy is a metric used in information theory that evaluates the impurity or uncertainty in a set of data. It controls how a decision tree divides data. The value of xi and yi is the probability of p(xi,yi) shown in equation (2).

$$H(X,Y) = -\sum_{xi \in X} \sum p(x_i, y_i) \log p(x_i, y_i) \quad (2)$$

D. Convolutional Neural Network (CNN)

SSD mobilNetV2 have been load into CNN which is a pre-trained model for object detection. ReLU activation function and MaxPooling layers are added after the first Convolution layer. The Convolution layer is taught by hundreds of filters. Each of the filter is to determine the edge of the image object. The size of the kernel can be set into certain size based on height and width convolution of two dimensional. It should be aware of the predicted shape of the input, input shape information must be sent to the first layer of the model. Following layers are capable of conducting shape reckoning instinctively.

The Conv2D class's activation parameter is set to "relu." It depicts a nearly linear function with all of the benefits of linear models, including the ease with which gradient-descent methods may be used to optimize it. By choosing the Max Pooling size of 7 x 7, the spatial dimensions can be lowered. The next convolution layer has 100 filters, each of which is activated using the ReLU activation function and is followed by another MaxPooling layer.

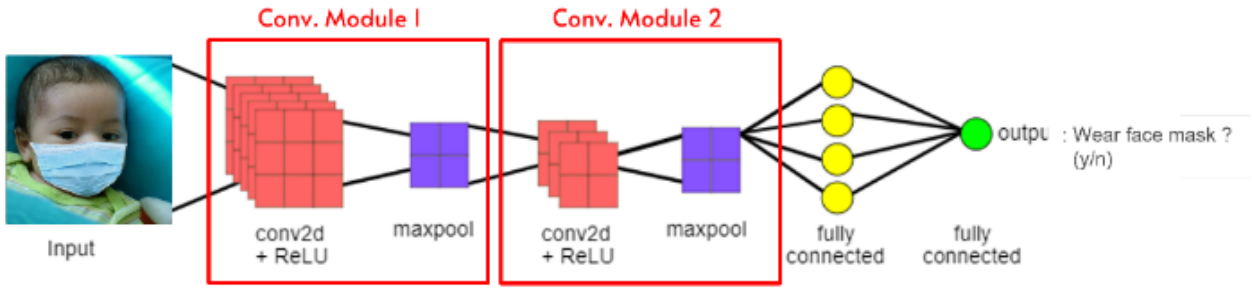

Figure 6 : Convolutional Neural Network architecture for the system

To avoid overfitting, a Dropout layer with a 50% probability of setting inputs to zero is added to the model. A Dense layer of 64 neurons with a ReLu activation function is then added. In the last layer (Dense), which has two outputs for two categories, the Softmax activation function is applied.

The compile method must first be used to configure the learning process. The "adam" optimizer is used here. As a loss function, categorical cross entropy, also known as multiclass log loss, is employed (the objective that the model tries to minimize). Because this is a classification problem, the metrics are set to "accuracy".

E. Split the data into train and test

When generating a prediction, having a good model and a well-designed train test split can help to get accurate results. The test size is set at 0.1, implying that 90% of the data in the dataset is utilized for training and 10% for testing. The sequential is then fitted to the pictures in the training and test sets. In this project, 20% of the training data is used for validation. The model is trained for 20 epochs (iterations) in order to strike a compromise between accuracy and overfitting risk. This will consist of what it thinks that the image is either people wearing facemask, wearing mask incorrectly or not wearing a mask. It will actually consist of probabilities, such as 70% for wearing facemask, 20% for not wearing and 10% for wearing mask incorrectly. This is where the error need to be reduced.

3.5 Calculation of Performance

In each model algorithm, performance testing has been calculate using performance criteria employed include accuracy, precision, recall, and F1 score by referring to the confusion matrix. The best performing parameter in one algorithm will be chosen and compared to another model.

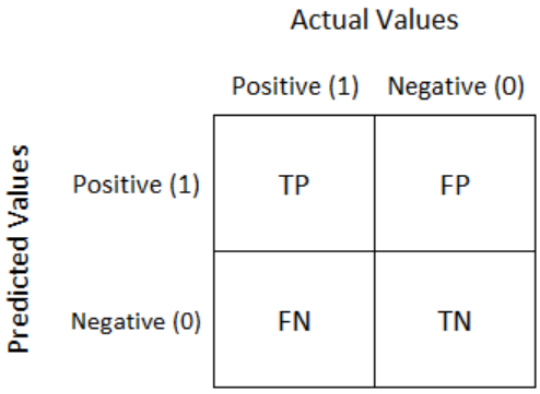

Figure 7 : Confusion Matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{3}$$

Accuracy values is calculated using Equation (3). TP stand for true positive, and FP stand for false positive which mean this is a positive class population while TN and FN stand for true negative and false negative respectively which include them in negative class population[9]. Then to calculate the precision is by using Equation (4).

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

After that, next evaluation is Recall. The proportion of truth of the positive class prediction relative to the entire data with an actual positive label is used to compute Recall and the calculation can be compute using Equation (5).

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

And lastly to calculate the F1-score is by using Equation (6).

$$F1 - score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad (6)$$

All this equation are included in the SKlearn library package Python. Just use the function of classification report and the result will appear. But certain model will require certain time to train and testing.

4 RESULTS AND DISCUSSION

This is the performance value of each of the model after train and testing the dataset of three different class which is wear face mask, wear face mask incorrect and not wearing face mask by also showing confusion matrix and the value of accuracy, precision, recall and F1-score by using the calculation method that have been proposed before.

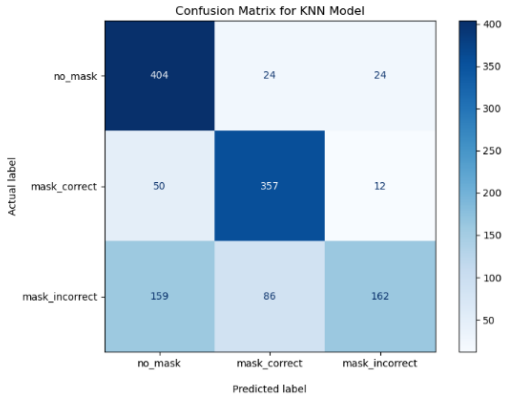

Figure 8 : KNN model 5 neighbour confusion matrix

Table 1 : Accuracy model of 5 neighbour

Class	Precision	Recall	F1 score
Wear mask incorrect	0.82	0.4	0.54
Wearing a mask	0.76	0.85	0.81
Not Wearing a mask	0.66	0.89	0.67
Accuracy	0.72		

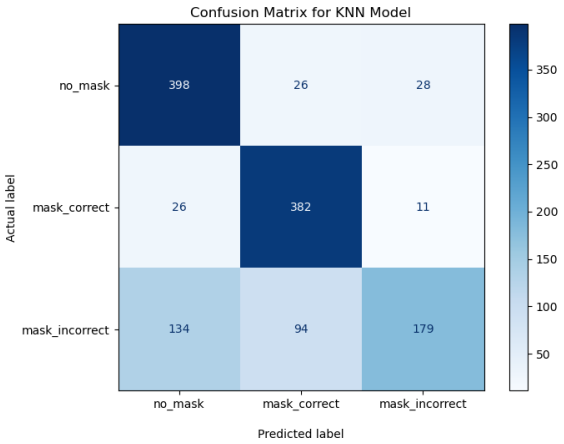

Figure 9 : KNN model 7 neighbour confusion matrix

Table 2 : Accuracy model of 7 neighbour

Class	Precision	Recall	F1 score
Wear mask incorrect	0.82	0.44	0.57
Wearing a mask	0.76	0.91	0.83
Not Wearing a mask	0.71	0.88	0.79
Accuracy	0.75		

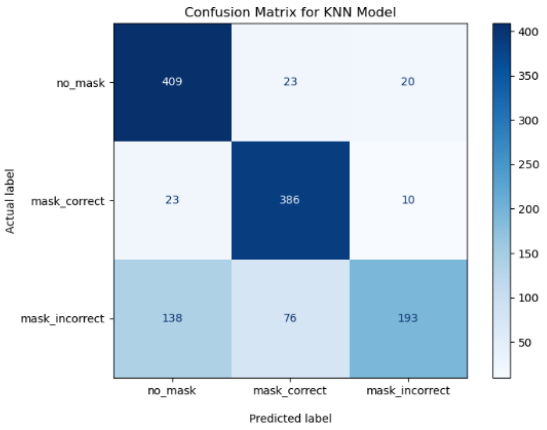

Figure 10 : KNN model 9 neigbour confusion matrix

Table 3 : Accuracy model of 9 neighbour

Class	Precision	Recall	F1 score
Wear mask incorrect	0.87	0.47	0.61
Wearing a mask	0.8	0.92	0.85
Not Wearing a mask	0.72	0.9	0.8
Accuracy	0.77		

Table 4 : Accuracy of SVM model

Class	Precision	Recall	F1 score
Wear mask incorrect	0.68	0.68	0.68
Wearing a mask	0.93	0.96	0.95
Not Wearing a mask	0.5	0.11	0.68
Accuracy	0.9		

Table 5 : Accuracy of Decision Tree model

Class	Precision	Recall	F1 score
Wear mask incorrect	0.83	0.97	0.83
Wearing a mask	0.92	0.98	0.93
Not Wearing a mask	0.87	0.85	0.86
Accuracy	0.87		

Table 6 : Accuracy of CNN model

Class	Precision	Recall	F1 score
Wear mask incorrect	0.94	0.97	0.95
Wearing a mask	0.99	0.98	0.99
Not Wearing a mask	0.98	0.95	0.97
Accuracy	0.97		

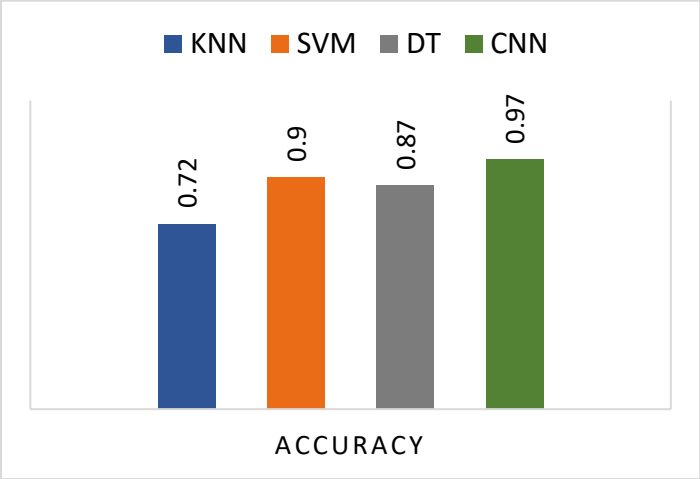

Figure 11 : Accuracy of each model

By refer to the result, CNN have the higher model accuracy compared to others with 97% accuracy while KNN give the lowest accuracy with 72% but with the value of neighbours increase the accuracy also increase. So, CNN are the best model to this system. Figure 24 shown the flow diagram of the system using CNN model.

After training, validating, and testing the model on the datasets, the technique achieves up to 99 percent accuracy for three separate classes. One of the key reasons for obtaining this degree of precision is MaxPooling. It provides basic internal representation while minimising the number of parameters that the model must learn. By down sampling the input picture representation, this sample-based discretization approach decreases its dimensionality. The initial epoch set for this training is 20 but with the error reduced close to zero value, it stops at epoch 13. The main purpose for training the model is to reduce the error close to zero as possible. By increase the number of epoch iteration, the error will be reduced, and the accuracy of training will increase.

When cover a face with mask or a hand, the system can recognise partially obscured faces with excellent accuracy. To eliminate between annotated mask and hand-covered face, it examines the degree of occlusion of four places – nose, mouth, chin, and eye. As a result, the model will only regard a mask that totally covers the face, including the nose and chin, as "with mask." The method's key issues include shifting perspectives and a lack of clarity. The unclear shifting faces in the live feed make it much more challenging.

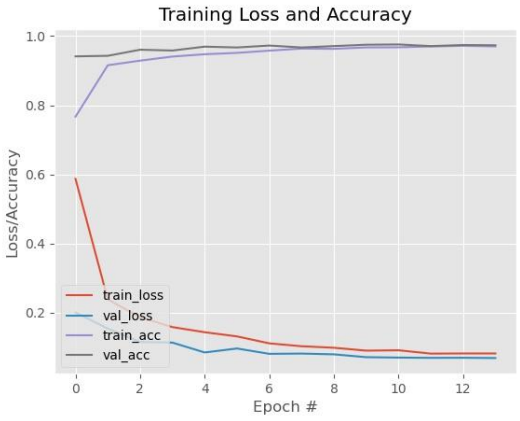

Figure 12 : Training result for 3 different classes

Figure 13 : Training result for 2 different classes

For 2 different classes, same initial epoch set with 3 different classes (20 epoch). The iteration stops at epoch 6 because the error has been reduced to very minimum value.

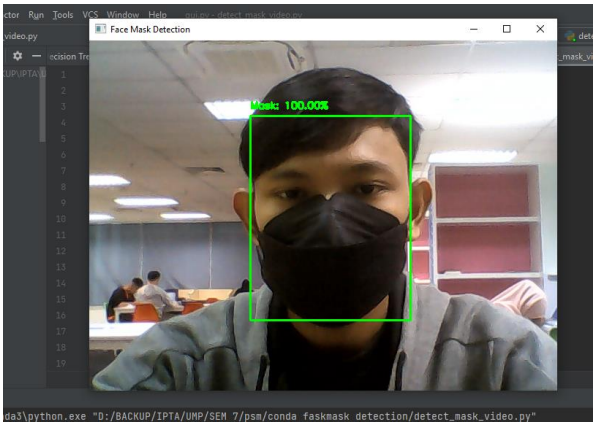

Figure 14 : Face mask detect

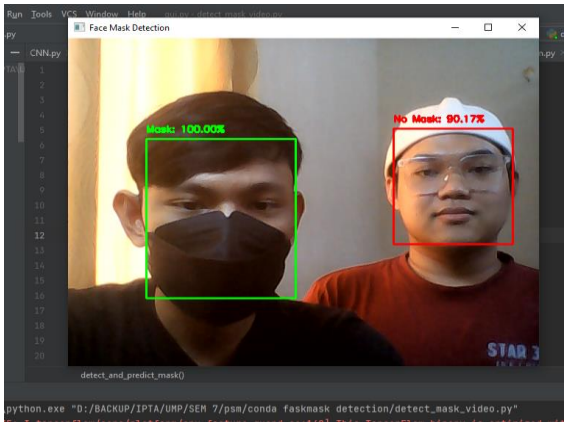

Figure 15 : Face mask detect with 2 person in camera frame

For KNN model, the value of neighbour can be manipulated to increase the accuracy of model, the time compute for the model to compute is 1.78 seconds for 5 neighbour, 2 seconds for 7 neighbour and lastly 1.85 seconds for 9 neighbours. Here to remember that KNN cannot read large amount of data because it will lead to noise of the data. That comes with SVM model to overcome the drawback of KNN. Face detector are used to detect the face of people within the camera frame. The system could detect faced based on the brightness of the video.

5 CONCLUSION

As a conclusion, the artificial intelligence techniques known as the techniques that were used increasingly to model environmental systems. The techniques covered are machine learning, deep learning, artificial neural networks, convolutional neural network and also supervised learning. This algorithm is well-suited to serve as a face detector in a face mask detection system before displaying the final result of whether or not people are wearing masks. The Python language, in conjunction with the Pycharm software, proved to be extremely useful, as it handled all of the process and because of it have a huge community support, to make an algorithm is easy by referring to many open sources site. Artificial intelligence is excellent for a wide range of system development, particularly image detection and recognition, but it requires additional training and troubleshooting to achieve high accuracy in the final result. The application created with the proper model will undoubtedly make all users accessible and everyone's life more convenient.

ACKNOWLEDGEMENT

I'd like to express my gratitude to Universiti Malaysia Pahang and Prof. Madya Ir. TS. Dr Fahmi Bin Samsuri, my supervisor, for his guidance and expertise in completing this project. In addition, I'd want to convey my heartfelt gratitude to my parents for their unwavering spiritual support and encouragement in helping me complete this project. Their thoughts and prayers keep me going and make me believe that I can achieve it. I've also recognized that performing this project will polish my skills. Finally, I'd like to express my gratitude to all my friends that assisted me in completing this project. I am grateful for all your assistance.

REFERENCES

[1] X. Fan and M. Jiang, “RetinaFaceMask: A Single Stage Face Mask Detector for Assisting Control of the COVID-19 Pandemic,” Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern., pp. 832–837, 2021, doi: 10.1109/SMC52423.2021.9659271.

[2] Y. Chung, C.-A. Chou, and C.-Y. Li, “Central Attention and a Dual Path Convolutional Neural Network in Real-World Tree Species Recognition,” Int. J. Environ. Res. Public Health, vol. 18, no. 3, 2021, doi: 10.3390/ijerph18030961.

[3] Y. Chen et al., “Face Mask Assistant: Detection of Face Mask Service Stage Based on Mobile Phone,” IEEE Sens. J., vol. 21, no. 9, pp. 11084–11093, 2021, doi: 10.1109/JSEN.2021.3061178.

[4] S. Ghosh, N. Das, and M. Nasipuri, “Reshaping Inputs for Convolutional Neural Network: Some Common and Uncommon Methods,” Pattern Recogn., vol. 93, no. C, pp. 79–94, Sep. 2019, doi: 10.1016/j.patcog.2019.04.009.

[5] B. Suvarnamukhi and M. Seshashayee, “Big Data Concepts and Techniques in Data Processing,” Int. J. Comput. Sci. Eng., vol. 6, no. 10, pp. 712–714, 2018, doi: 10.26438/ijcse/v6i10.712714.

[6] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau, “Visual Analytics in Deep Learning: An

Interrogative Survey for the Next Frontiers," IEEE Trans. Vis. Comput. Graph., vol. 25, no. 8, pp. 2674–2693, 2019, doi: 10.1109/TVCG.2018.2843369.

[7] C. Kanan and G. W. Cottrell, "Color-to-grayscale: Does the method matter in image recognition?," PLoS One, vol. 7, no. 1, 2012, doi: 10.1371/journal.pone.0029740.

[8] M. Hashemi, "Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation," J. Big Data, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0263-7.

[9] M. Nur and Y. Utomo, "Face Mask-e Wearing Detection Using Soft-Margin Support Vector Machine (SVM)," vol. 10, no. 2, pp. 72–81, 2021, doi: 10.14421/ijid.2021.3038.

[10] A. Malge, H. M. Dhaduk, and P. M. Mallikarjuna Shastry, "An approach to face detection and recognition using viola jones," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 5 Special Issue, pp. 52–56, 2019.

[11] I. Gogul and V. S. Kumar, "Flower species recognition system using convolution neural networks and transfer learning," *2017 4th Int. Conf. Signal Process. Commun. Networking, ICSCN 2017*, no. March, 2017, doi: 10.1109/ICSCN.2017.8085675.

[12] M. Abadi *et al.*, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," 2016, [Online]. Available: http://arxiv.org/abs/1603.04467.

[13] M. M. Rahman, M. M. H. Manik, M. M. Islam, S. Mahmud, and J. H. Kim, "An automated system to limit COVID-19 using facial mask detection in smart city network," *IEMTRONICS 2020 - Int. IOT, Electron. Mechatronics Conf. Proc.*, 2020, doi: 10.1109/IEMTRONICS51293.2020.9216386.

[14] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, *A survey of the recent architectures of deep convolutional neural networks*, vol. 53, no. 8. Springer Netherlands, 2020.

[15] A. Oumina, N. El Makhfi, and M. Hamdi, "Control the COVID-19 Pandemic: Face Mask Detection Using Transfer Learning," *2020 IEEE 2nd Int. Conf. Electron. Control. Optim. Comput. Sci. ICECOCS 2020*, pp. 19–23, 2020, doi: 10.1109/ICECOCS50124.2020.9314511.