Face Mask-Wearing Classification using Machine Learning

Introduction

In late December 2019, a seafood wholesale wet market in Wuhan, Hubei, China, experienced an outbreak of a strange pneumonia characterised by fever, dry cough, weakness, and occasional gastrointestinal symptoms (Huang et al., 2020). The pathogen of the outbreak was later identified as a novel beta-coronavirus named 2019 novel coronavirus (Wu et al., 2020). World Health Organization (WHO) officially named the disease as Coronavirus Disease-2019 (COVID-19). COVID-19 is a respiratory illness that causes severe pneumonia in those who are infected. As the disease spreads to practically every country on the planet, the majority of the world's population has been affected. WHO revealed that there were 468 million confirmed cases of COVID-19 and 6 million COVID-19 related deaths worldwide as of 20 March 2022 (WHO, 2022). The virus can enter the host through the respiratory system or mucosal surfaces like the conjunctiva. Therefore, COVID-19 is spread through salivation beads, respiratory droplets, and nasal droplets released when an infected person coughs, sneezes, or breathes the virus into the atmosphere.

Since there are no known treatments for COVID-19, it is indeed critical to avoid infection and transmission. The spread of COVID−19 can be limited if people strictly follow the standard operating procedures (SOPs), such as maintaining social distancing and wearing a face mask. As of March 2022, wearing of face mask remained mandatory in public areas in Malaysia (Carvalho & Kaos, 2022). Although people are wearing face masks, some of them are not wearing them correctly. For example, some people tend to wear them under the nose, on the tip of the nose, or folded above the chin. Detecting the people not obeying the mandatory mask-wearing rules and informing the corresponding authorities can be a solution in reducing the spread of COVID−19. To minimize the spread of COVID-19 in public places such as shopping malls and schools, security officer(s) is often needed at the entrances to check if each visitor is wearing a face mask. However, manual detection of the visitors not obeying the rules can be a difficult and labour-intensive task. It is also challenging for a security officer to detect the visitors who are not wearing their face masks correctly. To make sure people are wearing masks properly and correctly, an effective and efficient computer vision and machine learning strategy is required. Such techniques can be implemented in automatic face mask detection system, which can be installed at the entrances of public areas to identify people who are not wearing face masks correctly, in addition to those who are not wearing face masks at all. The automatic face mask detection system is reliably more accurate and faster than traditional manual detection using manpower. Since it is able to replace the need of manpower, it is also more cost-effective in the long run.

Image classification, which is a big part of machine learning, is a process in computer vision which classifies images based on their visual content and predefined categories (Naufal et al., 2021). Deep learning is very often used in the case of face mask detection due to its high level of accuracy. Several studies have shown that Convolutional Neural Networks (CNN); for instance, VGG-16, Resnet, MobileNet, are efficient in face mask detection (Goyal et al., 2022). However, these models often require large memory and computational time. The challenge of the face mask detection system is not only about achieving high accuracy, but also having computational efficiency so that it can be implemented easily and inexpensively with minimum resource requirement in various public places. Therefore, more research into accurate and computationally efficient face mask identification algorithms is required.

This project proposes multiple classification machine learning models to identify and classify the different ways to wear a face mask. These models are Naïve Bayes (NB), Support Vector Machines (SVM), Decision Tree (DT), Random Forest (RF) and K-Nearest Neighbors (KNN).

Naive Bayes (NB)

Naïve Bayes is a probabilistic machine learning algorithm based on the Bayes Theorem, used in a wide variety of classification tasks. Bayes' Theorem is a simple mathematical formula used for calculating conditional probabilities. Conditional probability is a measure of the probability of an event occurring given that another event has (by assumption, presumption, assertion, or evidence) occurred. The fundamental Naïve Bayes assumption is that each feature makes a independent and equal contribution to the outcome (Chauhan, 2022).

Support Vector Machines (SVM)

The SVM algorithm can classify both linear and non-linear data. Each data item is first mapped onto an n-dimensional feature space, with n denoting the number of features. After that, the hyperplane that divides the data into two groups is found, with the marginal distance for both classes maximised and classification errors minimised. The marginal distance between the decision hyperplane and its nearest instance, which is a member of a class, is the marginal distance for that class. Each data point is first plotted as a point in an n-dimensional space (where n is the number of features), with the value of each feature equal to the coordinate value (Uddin et al., 2019).

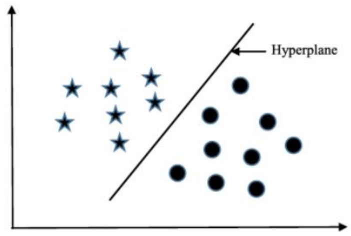

Figure 1 *Support Vector Machines (Uddin et al., 2019)*

Decision Tree (DT)

DT is one of the first and most well-known machine learning techniques. DT represents the decision logics for classifying data objects into a tree-like structure, i.e., tests and outcomes. A DT tree's nodes usually have numerous layers, with the root node being the first or top-most node. All internal nodes (those with at least one child) reflect input variable or attribute testing. The classification algorithm branches towards the appropriate child node based on the test result, and the process of testing and branching repeats until it reaches the leaf node. The choice outcomes are represented by the leaf or terminal nodes. DTs are a common component of many medical diagnostic regimens since they are simple to understand and learn. When traversing the tree for the classification of a sample, the outcomes of all tests at each node along the path will provide sufficient information to conjecture about its class.

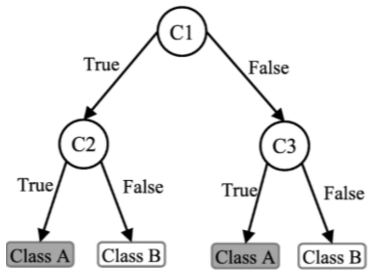

Figure 2 *Decision Tree (Uddin et al., 2019)*

Random Forest (RF)

RF is a multi-Decision Tree (DT) ensemble classifier, similar to how a forest is made up of many trees. Deep DTs are prone to overfitting the training data, producing a large variance in classification results for a minor change in the input data. The several DTs of an RF are trained using distinct parts of the training dataset. The sample's input vector must be handed down with each DT of the forest to categorise a new sample. Following that, each DT considers a different segment of the input vector to arrive at a categorization decision. The forest then determines whether to adopt the classification with the most 'votes' (for discrete classification outcomes). The RF algorithm can reduce the variance generated by merely evaluating one DT for the same dataset as it takes into account the results of numerous separate DTs (Uddin et al., 2019).

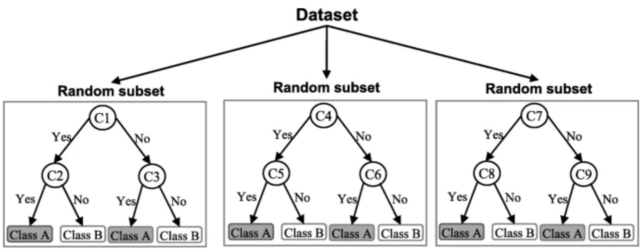

Figure *3 Random Forest (Uddin et al., 2019)*

K-Nearest Neighbor (KNN)

One of the simplest and earliest classification techniques is the KNN algorithm. The number of nearest neighbours considered to take a 'vote' is the 'K' in the KNN algorithm. For the same sample object, various values for 'K' can result in different classification results.

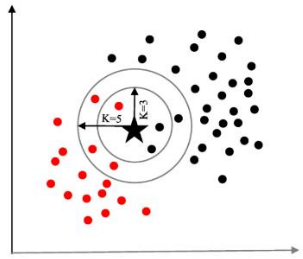

Figure 4 *K-Nearest Neighbors (Uddin et al., 2019)*

Objectives

1. To perform Exploratory Data Analysis (EDA) on the image dataset.
2. To implement supervised machine learning models for face mask-wearing image classification.
3. To evaluate the performance of each model using evaluation metrics.

Literature Review

Utomo et al. (2021) implemented a face mask-wearing detection system using SVM. The dataset used was obtained from Kaggle, namely the Face Mask Detection (FMD) dataset. This dataset had 1915 images with face mask and 1918 images without face mask. Feature selection in this research was conducted by cropping the face area from a complete image to reduce features and leave only an important part of the region of interest (ROI). Haar Cascades Frontal Face was utilized to detect facial regions in an image. Then, OpenCV was used to capture/crop the face from the full image. Image pre-processing was performed in four stages, which were resizing the dimensions of the dataset image, transforming the image into a matrix of integers, reshaping the matrix into a list of arrays, and simplifying the array dimensions through a flattening process so it could be processed using SVM. The authors added a soft-margin objective to the SVM to overcome images that cannot be separated linearly both in the training process and in the implementation stage. The proposed model was evaluated in terms of accuracy, precision, recall and F-measure. The evaluation was performed using a confusion matrix with 10-folds cross validation. This research also compared computational efficiency in terms of training and classification speed between SVM and the state-of-the-art method, CNN. As a result, CNN had a higher accuracy than SVM. However, the training speed of CNN was much lower than that of SVM. CNN also required a longer time than SVM in terms of classifying face masks on a new image.

Chugh et al. (2020) conducted a comparative analysis on image classification using KNN, RF and Multi-Layered Perceptron (MLP), which is a feed forward neural network. The dataset used was the Fashion-MNIST dataset, encompassing over 70000 images belonging to 10 classes. A feature optimization technique, namely Principal Component Analysis (PCA), was carried out to reduce the spatial dimensions of feature space. Grid Search, which is a process of implementing hyperparameter tuning to acquire the optimal values of a given model, was performed on the models to improve their accuracies. The results showed that MLP model yielded the best accuracy, followed by RF and then KNN. On the other hand, The RF model was the most time efficient of the three, followed by KNN and MLP. Because of the great number of computations that had to be done repeatedly for each epoch, the MLP model had a greater temporal complexity than the other models.

In the research performed by Kumar et al. (2012), KNN, DT and SVM were implemented for image classification using a general-purpose image database containing 500 JPEG images from COREL photo gallery. Histogram extraction was utilized for colour features extraction. The evaluation metrics used to assess the performance of each model were accuracy, precision, and recall. From the results, it was observed that SVM had the best performance, despite having suffered from features outlier problem. The authors mentioned that such problem could be removed through optimization of data processing technique using genetic algorithm.

Vijitkunsawat et al. (2020) studied the performance of the three algorithms: KNN, SVM and MobileNet to identify the best algorithm for real-time face mask detection. The dataset comprised of 690 images with face masks and 686 images without face masks. Data augmentation was implemented to increase the size of the dataset by rotating, flipping and blurring the images. Accuracy was the only evaluation metric adopted by the authors. From the experiment, MobileNet showed the highest efficiency compared to KNN and SVM.

WEKA software was used by Jankovic et al. (2019) in their study, which evaluated the performance of multiple decision tree classifiers for image classification. A small sample of cultural heritage image dataset was used in this study. The authors stated that using an open-source software, such as WEKA, for image classification lowered the costs of such a task. Three types of extracted image features were used, namely Fuzzy and texture histogram, edge histogram, and DCT coefficients. The extraction of the features was made in Weka, using the ImageFilter package which is based on LIRE, a Java library for image retrieval. Based on the extracted features, J48, Hoeffding Tree, Random Tree, and RF were applied on the dataset for classification. The models were tested using 10-fold cross validation. To evaluate the performance of each model, percent of correctly classified instances, Kappa statistics, Mean Absolute Error (MAE), precision, recall, F-measure, and time taken to build the model in seconds were recorded. The obtained results indicated that RF had the best performance in terms of classification accuracy.

Methodology

In this project, a total of 1222 colour images (selfies of volunteers wearing their face masks in various forms) open-source by Marceddu et al. (2021) are used to build machine learning models for the classification of three different mask-wearing ways. All the images used in this project are pre-labelled by the authors into 8 classes: mask correctly worn, mask not worn, mask under the chin, mask hanging on an ear, mask under the nose, mask on the tip of the nose, mask on the forehead, and mask folded above the chin. The *Python* programming language and its libraries such as *OpenCV* (*OpenCV*, 2022), *TensorFlow* (*TensorFlow*, 2022), and *Scikit-Learn* (*Scikit-Learn*, 2022) are used for the implementation of the models.

Data Understanding

Exploratory Data Analysis (EDA) is carried out to examine the characteristics and structure of the images so that the image pre-processing techniques needed could be identified and performed in the next phase. *Matplotlib* and *Pillow* libraries are used to visualize the number of images assigned to each class and the raw image sizes for each class.

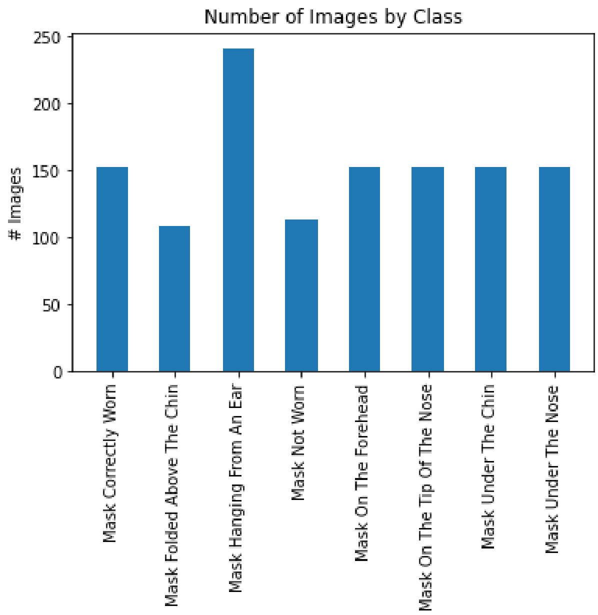

Figure 5 *Bar chart that shows the number of images in each class.*

From the bar chart generated above, it can be seen that most classes have about the same number of images, which is around 150. The “mask hanging from an ear” class has about 245 images, while “mask folded above the chin” and “mask not worn” classes have about 110 images each.

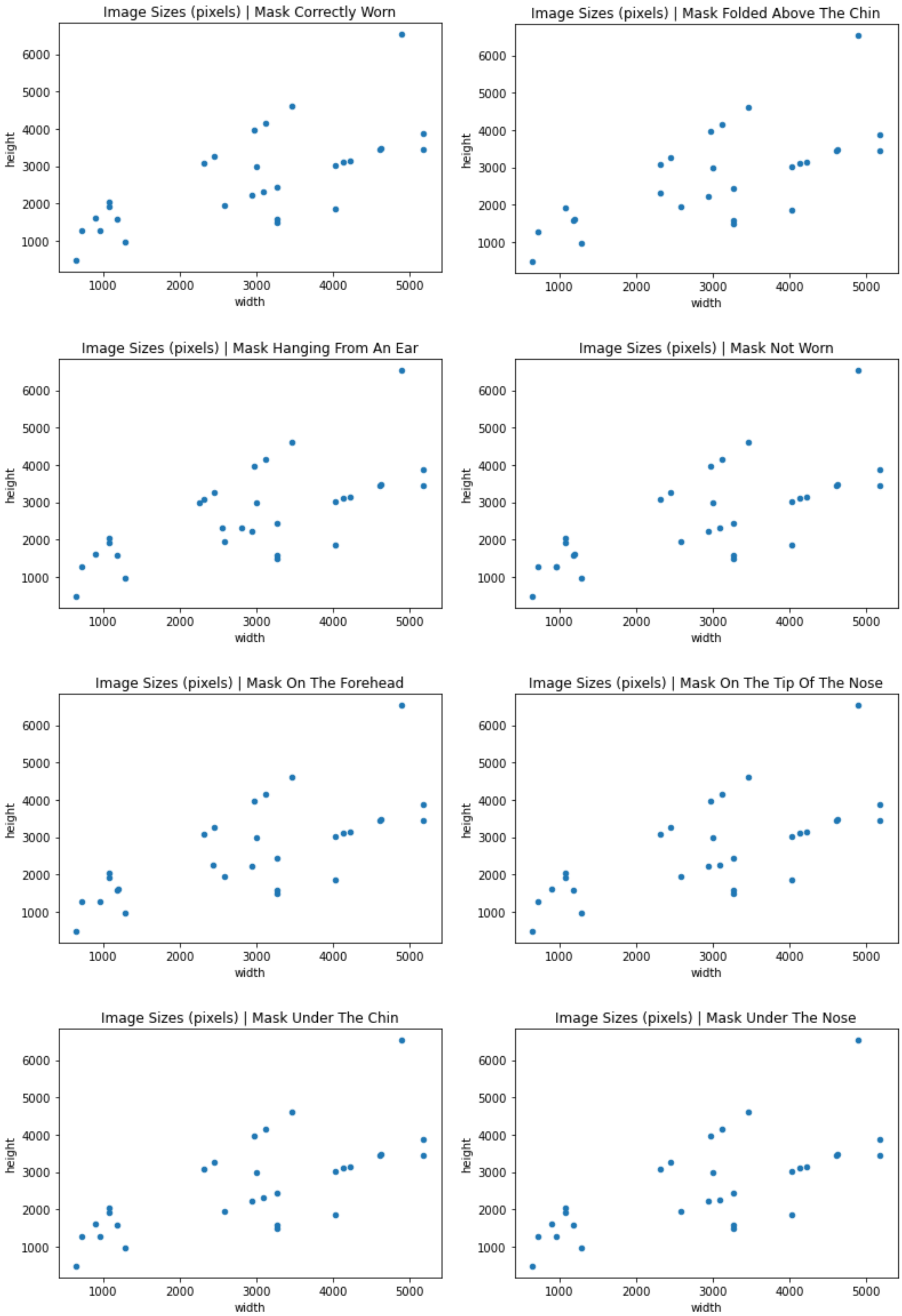

Figure 6 *Scatter plots that show the resolutions of images in each class.*

The scatter plots above show that the distribution of image sizes in pixel is similar across all the classes. However, most of the images' resolutions are not the same and their resolutions are huge, with an average of *2827 x 3592*. All of the images are colour image, so there are a total of three channels (i.e., three planes of *width x height* resolution) in each of the images. Each pixel of a channel has one possible value range from 0 – 255 to represent the intensity of the colour that constitute that pixel.

Image Pre-processing

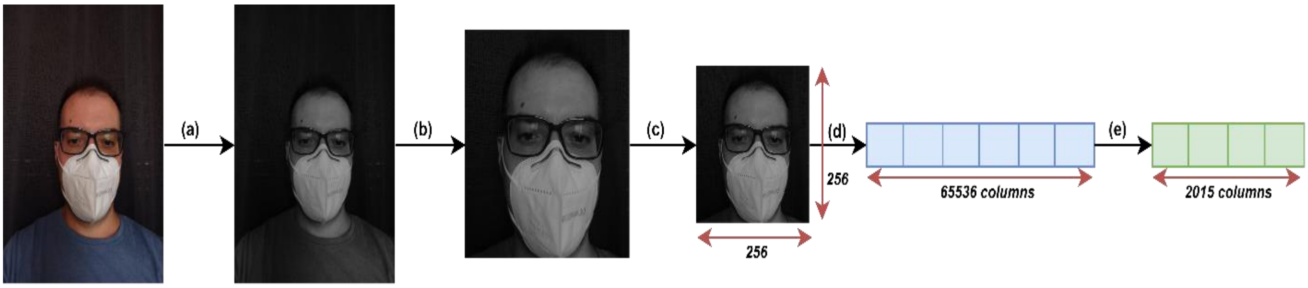

Figure 7 *The image pre-processing steps.*

(a) Convert colour images to grayscale
All the colour images (from three channels) are converted to grayscale (to one channel) using *OpenCV* library as colour information such as mask colour and skin colour are redundant in the mask wearing ways detection use case and feeding the colour information to the models might result in a model that is unable to detect the mask-wearing way correctly if a person is wearing a mask with colour that is not available in the training images. Furthermore, triple of the processing capacity is required to work with three-channels colour image as compared to one-channel grayscale image.

(b) Retain only the central region of image
All the grayscale images are cropped with the fraction of 0.5 (50%) to retain only the central region using *TensorFlow*. In an image, the region of interest is the central region that occupies with a person face. Retaining only the region of interest will help the models in capturing important patterns and eliminates the noise introduced by the irrelevant outer region.

(c) Image resizing
In the data understanding phase, we found that most of the images' resolutions are not the same and their resolutions are huge. However, machine learning models can only receive inputs of the same size and the larger the image resolution size, the larger memory and processing capacity are needed to process the images. Hence, image resize is needed to scale down the images to resolution recommended for image classification model training, which is *256 x 256* resolution (Sabottke & Spieler, 2020).

(d) Image flattening as model inputs
The machine learning models require the input feed in the form of one-dimensional array. Hence, all the scaled down images are flattened from two-dimension to one-dimension. A two-dimensional image with resolution of *256 x 256* is now in one-dimension with size of 65,536 columns. A single image is represented by a single one-dimension array with each column represents a single input feature. This is analogous to the use of structured dataset where each row represents a single record, and each column represents a single field of the record.

(e) Inputs dimensionality reduction
The input features that we obtained for a single image is very huge (65,536 columns) even though the images were cropped and scaled down. The input data with too many features is likely to overfit the machine learning models on the training data by capturing the noise or irrelevant information within the data, thereby causing the model to perform poorly on the testing data. To prevent this issue, dimensionality reduction technique – Kernel Principal Components Analysis (KPCA) with Radial Basis Function, which is an extension of PCA that applies non-linear transformation, is used to reduce the number of input features of all the images. This technique transforms large input features into smaller number of principal components as new input features, where each principal component represents a percentage of the total variance captured from the data. All the inputs features values are first standardized to the range of 0 – 1 so that the feature with higher pixel intensity value will not dominate over features with smaller pixels values. The dimensionality reduction technique is then applied on the standardized inputs features, which results in a total number of 1220 principal components or features for each image.

Data Preparation

As we are only interested in detecting the people who are wearing the mask correctly, not wearing a mask, or wearing the mask incorrectly. All the images in mask under the chin, mask hanging on an ear, mask under the nose, mask on the tip of the nose, mask on the forehead, and mask folded above the chin classes; are merged into a single class called mask incorrectly worn, because all images represent incorrect ways of mask wearing and merging the classes can reduce the complexity of the models. The total number of images in mask correctly worn, mask not worn, and mask incorrectly worn classes are now 152, 113, and 957 respectively. The samples number in the first two classes are lower than the mask incorrectly worn class. So, image augmentation technique – horizontal and vertical image flipping, from *OpenCV* library, is used to up-sample the images in the two classes. After up-sampling, the total number of images in mask correctly worn, mask not worn, and mask incorrectly worn classes are now 608, 452, and 957 respectively. K-fold cross validation is used to divide the data into 10 folds to ensure that different portions of the data are used for training and testing the model at different iterations.

Models Building

All the inputs features are served as the independent variables to predict the 3 different mask-wearing ways: mask correctly worn (0), mask incorrectly worn (1), and mask not worn (2). From the literature review, four supervised machine learning models suited for image classification are identified; they are Naïve Bayes, Support Vector Machines, Decision Tree, Random Forest, and K-Nearest neighbours. All these models are built using the *Scikit-Learn* library in *Python*.

Models Evaluation

To assess the performance of the models, evaluation metrics which include accuracy, precision, recall, and F1-score are used. Comparing the scores obtained by each model can help us find the best performing one.

Results and Discussion

Using 10-fold cross validation, we obtained 10 sets of model performance scores in terms of accuracy, precision, recall, and F1-score, on the testing data. Averaging the scores gives us the overall performance of our machine learning models.

Machine Learning Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Trained Time (second)
Naïve Bayes	47.4	22.6	47.4	30.6	3.26
Support Vector Machines (Linear kernel)	47.4	22.7	47.4	30.6	63.98
Decision Tree	**85.7**	**85.9**	**85.7**	**85.7**	50.79
Random Forest	83.8	83.9	83.8	83.5	86.62
K-Nearest Neighbours (K = 5)	31.6	10.2	31.6	15.4	**0.3**

Table 1 *The overall performance scores obtained by the models.*

Table 1 shows the overall scores obtained by Naïve Bayes, Support Vector Machines, Decision Tree, Random Forest, and K-nearest neighbour's models.

By referring to Table 1, we can observe clearly that Decision Tree model outperforms the other four models in this image classification task. The training time of K-nearest neighbour (KNN) is significantly lower than the rest of the models; however, it has the worst prediction performance, where the accuracy is 33.33 % lower than a random classifier (random pick). Support Vector Machines and Naïve Bayes performed better than KNN, but the accuracy and precision are still considerably low. The tree-based algorithms, which are Decision Tree and Random Forest, performed better in this image classification task. Both algorithms achieved more than 80% for accuracy, precision, recall and F1-score, which is considered a good model performance. Decision Tree has better results as compared to Random Forest. For all evaluation metrics, Decision Tree achieved around 2% higher scores than Random Forest. Furthermore, the time taken to train a Decision Tree (50.79 seconds) is lesser as compared to Random Forest (86.62 seconds). One possible reason that the performances of Naïve Bayes, Support Vector Machines, and K-Nearest Neighbours are worse than the tree-based algorithms is that our target classes are non-linearly separatable, so the non-linear Decision Tree and Random Forest models perform better.

Therefore, Decision Tree is the best model among these five models. The accuracy (%), precision (%), recall (%), F1-score (%) and trained time (seconds) achieved by Decision Tree are 85.7, 85.9, 85.7, 85.7 and 50.79, respectively. From the accuracy metric, it shows that on average 85.7% of the time Decision Tree is able to classify the images classes correctly. The precision score achieved shows that the on average 85.9% of the images classes predicted by Decision Tree are in fact having that particular class. In addition, it shows that false positive is low where less images were wrongly classified. Next, the Decision Tree's recall shows that on average 85.7% of the images in each class were correctly identified by the model. Overall, the Decision Tree model has a good performance in image classification.

When looking through the images where the Decision Tree model predicted wrongly, we discovered that the model is able to predict the "mask under the nose" image correctly as "mask incorrectly worn", but wrongly predicted the "mask correctly worn" image as "mask incorrectly worn" (refer to Figure 8). This might be due to the similar patterns between these two images, where the incorrectly mask-wearing image on the left has the mask worn slightly below the nose, while the mask correctly worn image on the right has the mask fully covered the nose. So, the model is not able to distinguish the given patterns clearly.

Figure 8 *The mask under the nose (left) and mask correctly worn (right) images.*

To test whether the Decision Tree model can correctly predict unseen images, six mask-wearing images are randomly selected from Google (refer to Figure 9).

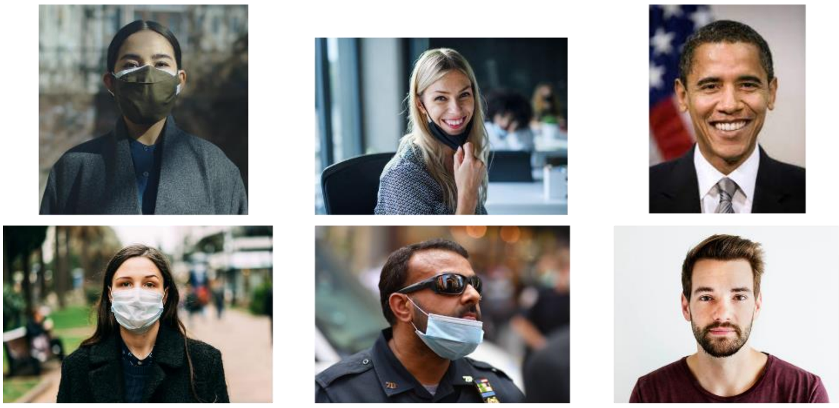

Figure 9 *The mask correctly wore (first column), mask incorrectly worn (second column), and mask not wore images (third column).*

After applied the same image pre-processing steps on the new images, Decision tree model is used to predict the images target class and the model predicted all the images as "mask incorrectly worn". This shows that the model is bias towards the "mask incorrectly worn" class, and this situation might be caused by the insufficient training samples for the "mask correctly worn" and "mask not worn" classes. Hence, the model comes up with the same prediction – "mask incorrectly worn", all the time.

Conclusion

In this project, we have identified and implemented five machine learning algorithms for face mask-wearing image classification task. Various pre-processing steps were applied on the dataset, which are colour image to grayscale conversion, retaining central region, image resizing, image flattening and dimension reduction. Four machine learning algorithms (Support Vector Classifier, K-Nearest Neighbours, Decision Tree and Random Forest) were used for face mask-wearing image classification. From the results, the worst to the best model are K-Nearest Neighbours, Support Vector Machine, Random Forest and Decision Tree. The Decision Tree model is the best among all the models where it achieved accuracy (%), precision (%), recall (%), F1-score (%) and trained time (seconds) of 85.7, 85.9, 85.7, 85.7 and 50.79, respectively.

When working on the implementation of models using *Python*, we encountered processing capacity limitation using personal laptop. The total size of our images is around 3.08 GB despite having small samples (1222 images is analogues to 1222 records in structured dataset) as most of the colour images are in high-resolution. Processing these high-resolution images require significant processor power to speed up the images processing time. By using laptop with 8GB RAM and 4 core processor with 2.4GHz, reading the images into *Python* alone took up to 9 minutes to finish execution. Nevertheless, we managed to achieve our objectives stated above despite having the processing capacity limitation.

For future works, we propose to increase the samples for “mask correctly worn” and “mask not worn” classes so that the models have enough samples to learn to distinguish between the classes. However, to handle more images, more processing power is needed. To solve this issue, we propose to utilise cloud computing resources that can give us the sufficient processing power required to process large images faster, without having to purchase a new powerful computer. Lastly, in recent years, many scientists found out that deep learning methods such as Convolutional Neural Network (CNN) outperform machine learning techniques in image classification (Goyal et al., 2022). Hence, we propose to experiment with deep learning algorithms for face mask-wearing image classification task to examine if better performance can be achieved as compared to the four machine learning algorithms used in this project.

References

Antonio Marceddu, Renato Ferrero, Bartolomeo Montrucchio. (2021). Ways to Wear a Mask or a Respirator (WWMR-DB). IEEE Dataport. https://dx.doi.org/10.21227/8atn-gn55

Carvalho, M. & Kaos, J. (2022, March 24). *Covid-19: Wearing of face mask remains mandatory, says Khairy*. The Star. https://www.thestar.com.my/news/nation/2022/03/24/covid-19-wearing-of-face-mask-remains-mandatory-says-khairy

Chauhan, N. (2022, April 8). *Naïve Bayes Algorithm: Everything You Need to Know.* KDnuggets. Retrieved June 18, 2022, from https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html

Chugh, R. S., Bhatia, V., Khanna, K., & Bhatia, V. (2020, January). A comparative analysis of classifiers for image classification. In 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 248-253). IEEE.

Goyal, H., Sidana, K., Singh, C., Jain, A., & Jindal, S. (2022). A real time face mask detection system using convolutional neural network. Multimedia Tools and Applications, 1-17.

Huang, C., Wang, Y., Li, X., Ren, L., Zhao, J., Hu, Y., ... & Cao, B. (2020). Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. The lancet, 395(10223), 497-506.

Jankovic, R. (2019, January). Classifying cultural heritage images by using decision tree classifiers in WEKA. In Proceedings of the 1st International Workshop on Visual Pattern Extraction and Recognition for Cultural Heritage Understanding Co-Located with 15th Italian Research Conference on Digital Libraries (IRCDL 2019), Pisa, Italy (pp. 119-127).

Kumar, S., Khan, Z., & Jain, A. (2012). A review of content based image classification using machine learning approach. International Journal of Advanced Computer Research, 2(3), 55.

Naufal, M. F., Kusuma, S. F., Prayuska, Z. A., Yoshua, A. A., Lauwoto, Y. A., Dinata, N. S., & Sugiarto, D. (2021). Comparative Analysis of Image Classification Algorithms for Face Mask Detection. Journal of Information Systems Engineering and Business Intelligence, 7(1), 56-66.

OpenCV. (2022). https://opencv.org/

Sabottke, C. F., & Spieler, B. M. (2020). The effect of image resolution on deep learning in radiography. *Radiology: Artificial Intelligence*, *2*(1), e190015.

Scikit-Learn. (2022). https://scikit-learn.org/stable/

TensorFlow. (2022). https://www.tensorflow.org/learn

Uddin, S., Khan, A., Hossain, M. E., & Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. BMC medical informatics and decision making, 19(1), 1-16.

Utomo, M. N. Y., & Violita, F. (2021). Face Mask Wearing Detection Using Support Vector Machine (SVM). IJID (International Journal on Informatics for Development), 10(2), 72-81.

Vijitkunsawat, W., & Chantngarm, P. (2020, October). Study of the performance of machine learning algorithms for face mask detection. In 2020-5th International Conference on Information Technology (InCIT) (pp. 39-43). IEEE.

World Health Organization. (2022, March 22). *Weekly epidemiological update on COVID-19 - 22 March 2022*. WHO. https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19---22-march-2022

Wu, Y. C., Chen, C. S., & Chan, Y. J. (2020). The outbreak of COVID-19: An overview. Journal of the Chinese medical association, 83(3), 217.