



Face Mask Wearing Classification Using Machine Learning [†]

Kalaimagal Ramakrishnan ¹, Vimala Balakrishnan ^{2,*}, Hui Yeok Wong ², Shi Hui Tay ², Kar Lok Soo ²
and Weng Kiat Kiew ²

¹ Faculty of Science and Engineering, University of Nottingham Malaysia, Semenyih 43500, Malaysia; kalaimagal.ramakrishnan@nottingham.edu.my

² Department of Information Systems, Faculty of Computer Science & Information Technology, Universiti Malaya, Wilayah Persekutuan Kuala Lumpur 50603, Malaysia

* Correspondence: vimala.balakrishnan@um.edu.my

[†] Presented at the International Conference on Electronics, Engineering Physics and Earth Science (EEPES'23), Kavala, Greece, 21–23 June 2023.

Abstract: In late December 2019, a cluster of previously unidentified coronavirus cases emerged in Wuhan, China. Subsequently, the virus quickly spread to the whole world in a matter of few months. At that point in time, there were no known treatments for COVID-19. Therefore, to limit the spread of virus transmission, the public was advised to maintain social distancing and wear a face mask. In Malaysia, most people were compliant and adhered to the standard of procedure (SOP). However, it was observed that many people were not wearing the mask correctly. Therefore, this paper aims to analyze how image classification using machine learning algorithms can be used to detect whether a face mask is properly worn. In this research, a total of 1222 color images (selfies) were used to build five machine learning models, in particular Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF) and K-Nearest Neighbors (KNN), to classify three methods of mask-wearing: mask correctly worn, mask incorrectly worn, and mask not worn. Our results show that Decision Tree is the best model among these five models in terms of accuracy (85.7%), precision (85.9%), recall (85.7%), and F1-Score (85.7%). However, it was observed that when classifying mask-wearing images the decision tree approach was not able to identify images with a similar pattern, for example, in the cases of mask under the nose and mask correctly worn. From an awareness perspective, this study emphasizes the need for the public to properly wear their face mask to reduce the spread of COVID-19 and the effectiveness of image classification in detection of face mask wearing.

Keywords: face mask detection; machine learning; image classification



Citation: Ramakrishnan, K.; Balakrishnan, V.; Wong, H.Y.; Tay, S.H.; Soo, K.L.; Kiew, W.K. Face Mask Wearing Classification Using Machine Learning. *Eng. Proc.* **2023**, *41*, 13. <https://doi.org/10.3390/engproc2023041013>

Academic Editor: Grigor Mihaylov

Published: 17 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In late December 2019, a seafood wholesale wet market in Wuhan, Hubei, China, experienced an outbreak of strange pneumonia characterized by fever, dry cough, weakness, and occasional gastrointestinal symptoms [1]. The pathogen of the outbreak was later identified as a novel beta-coronavirus named 2019 novel coronavirus [2]. The 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 has spread to practically every country on the planet, most of the world's population has been affected. The 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 [3]. The virus can enter the host through the respiratory system or mucosal surfaces such as 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. Because there were no known treatments for COVID-19, it was 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 a face mask remained mandatory in public areas in Malaysia [4]. Although people are wearing face masks, some of them are not wearing them correctly. For example, people tend to wear them under the nose, on the tip of the nose, or folded above the chin. Detecting people not obeying the mandatory mask-wearing rules and informing the corresponding authorities can be a solution to 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) are often needed at the entrances to check whether each visitor is wearing a face mask. However, manual detection of visitors not obeying the rules can be a difficult and labor-intensive task. It is challenging for a security officer to detect 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 an 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. Because it can replace the need for manpower, it is more cost-effective in the long run.

Image classification, which is a big part of machine learning, is a process in computer vision that classifies images based on their visual content and predefined categories [5]. 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, and MobileNet, are efficient in face mask detection [6]. 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, it requires sufficient computational efficiency to ensure that it can be implemented easily and inexpensively with minimum resource requirements in various public places. Therefore, more research into accurate and computationally efficient face mask identification algorithms is required. This paper proposes multiple machine learning classification models to identify and classify the different ways of wearing a face mask. These models are Naïve Bayes (NB), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbors (KNN). Naïve Bayes (NB) is a probabilistic machine learning algorithm based on the Bayes Theorem, and is 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 another event that has (by assumption, presumption, assertion, or evidence) occurred. The fundamental Naïve Bayes assumption is that each feature makes an independent and equal contribution to the outcome [7].

The Support Vector Machine (SVM) algorithm can classify both linear and nonlinear 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 maximized and classification errors minimized. 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 [8]. Decision Tree (DT) is one of the first and most well-known machine learning techniques. DT represents the decision logic used for classifying data objects into a tree-like structure, i.e., tests and outcomes. The nodes of a DT usually have numerous layers, with the root node being the first or topmost 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, as 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 provides sufficient information to make a conjecture about its class [9].

Random Forest (RF) is a multiple Decision Tree (DT) ensemble classifier, similar to how a forest is made up of many trees. Deep DTs are prone to overfitting to 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 in the forest to categorize a new sample. Next, 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 considers the results of numerous separate DTs [8]. One of the simplest and earliest classification techniques is the K-Nearest Neighbors (KNN) algorithm. The number of nearest neighbors 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 [10].

2. Related Works

Utomo et al. [11] implemented a face mask-wearing detection system using SVM. The dataset they used was obtained from Kaggle, namely, the Face Mask Detection (FMD) dataset. This dataset has 1915 images with face masks and 1918 images without face masks. 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 images. Then, OpenCV was used to capture/crop the face from the full image. Image preprocessing was performed in four stages: 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 could not be separated linearly in the training process or 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 ten-fold cross-validation. The authors compared the computational efficiency in terms of training and classification speed between SVM and a state-of-the-art CNN method. CNN had a higher accuracy than the SVM; however, the training speed of CNN was much lower than that of SVM, and CNN required a longer time than SVM to classify face masks on new images.

Chugh et al. [12] conducted a comparative analysis of 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 70,000 images belonging to ten classes. A feature optimization technique, namely, Principal Component Analysis (PCA), was carried out to reduce the spatial dimensions of the feature space. Grid search, which is a process for implementing hyperparameter tuning to acquire the optimal values of a given model, was performed on the models to improve their accuracy. The results showed that the MLP model yielded the best accuracy, followed by RF and then KNN. On the other hand, The RF model had the best time efficient of the three, followed by KNN and MLP. Because of the very large number of computations that had to be done repeatedly for each epoch, the MLP model had greater temporal complexity than the other models.

In research performed by Kumar et al. [13], KNN, DT, and SVM were implemented for image classification using a general-purpose image database containing 500 JPEG images from the COREL photo gallery. Histogram extraction was utilized for color feature 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 suffering from feature outlier problems. The authors mentioned that such problems could be removed through optimization with data processing techniques

using genetic algorithms. Vijitkunsawat et al. [14] 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 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. In their experiments, MobileNet showed the highest efficiency compared to KNN and SVM. WEKA software was used by Jankovic et al. [15] in their study, which evaluated the performance of multiple decision tree classifiers for image classification. A small sample of the cultural heritage image dataset was used in this study. The authors stated that using open-source software such as WEKA for image classification lowered the costs of the task. Three types of extracted image features were used, namely, fuzzy and texture histogram, edge histogram, and DCT coefficient. Feature extraction was carried out 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 to the dataset for classification. The models were tested using ten-fold cross-validation. To evaluate the performance of each model, the percentage of correctly classified instances, Kappa statistics, Mean Absolute Error (MAE), precision, recall, F measure, and time taken to build the model in seconds were all recorded. The obtained results indicated that RF had the best performance in terms of classification accuracy.

3. Materials and Methods

In this experiment, a total of 1222 open-source color images (selfies of volunteers wearing their face masks in various forms) provided by Marceddu et al. [16] were used to build machine learning models for the classification of three different mask-wearing states. All the images used in this paper were pre-labeled by the authors into eight 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 as OpenCV [17], TensorFlow [18], and Scikit-Learn libraries were used for the implementation of the models.

3.1. Data Understanding

Exploratory Data Analysis (EDA) was carried out to examine the characteristics and structure of the images so that the most applicable image preprocessing techniques could be identified and performed in the next phase. The Matplotlib and Pillow libraries were used to visualize the number of images assigned to each class as well as the raw image sizes for each class (Figures 1 and 2).

3.2. Image Preprocessing

The image preprocessing steps are presented as Figure 3 below.

1. Convert color images to grayscale;
2. Convert the color images (three channels) to grayscale (one channel) using the OpenCV library, as color information such as mask color and skin color are redundant in the mask-wearing detection use case, and feeding the color information to the models might result in a model that is unable to detect mask-wearing correctly if a person is wearing a mask with a color that is not available in the training images. Furthermore, three times as much processing capacity is required to work with a three-channel color images compared to one-channel grayscale images.

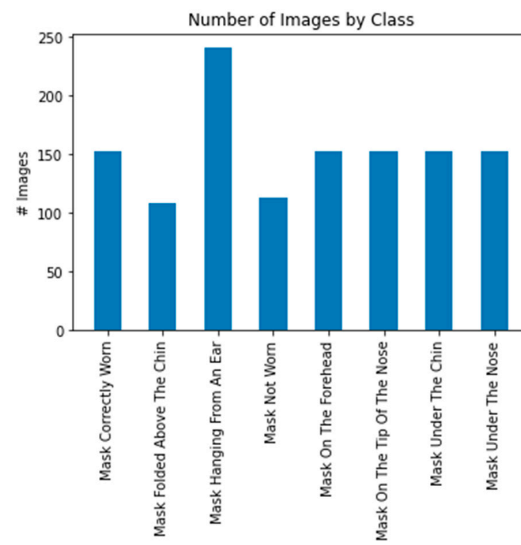


Figure 1. Bar chart showing the number of images in each class.

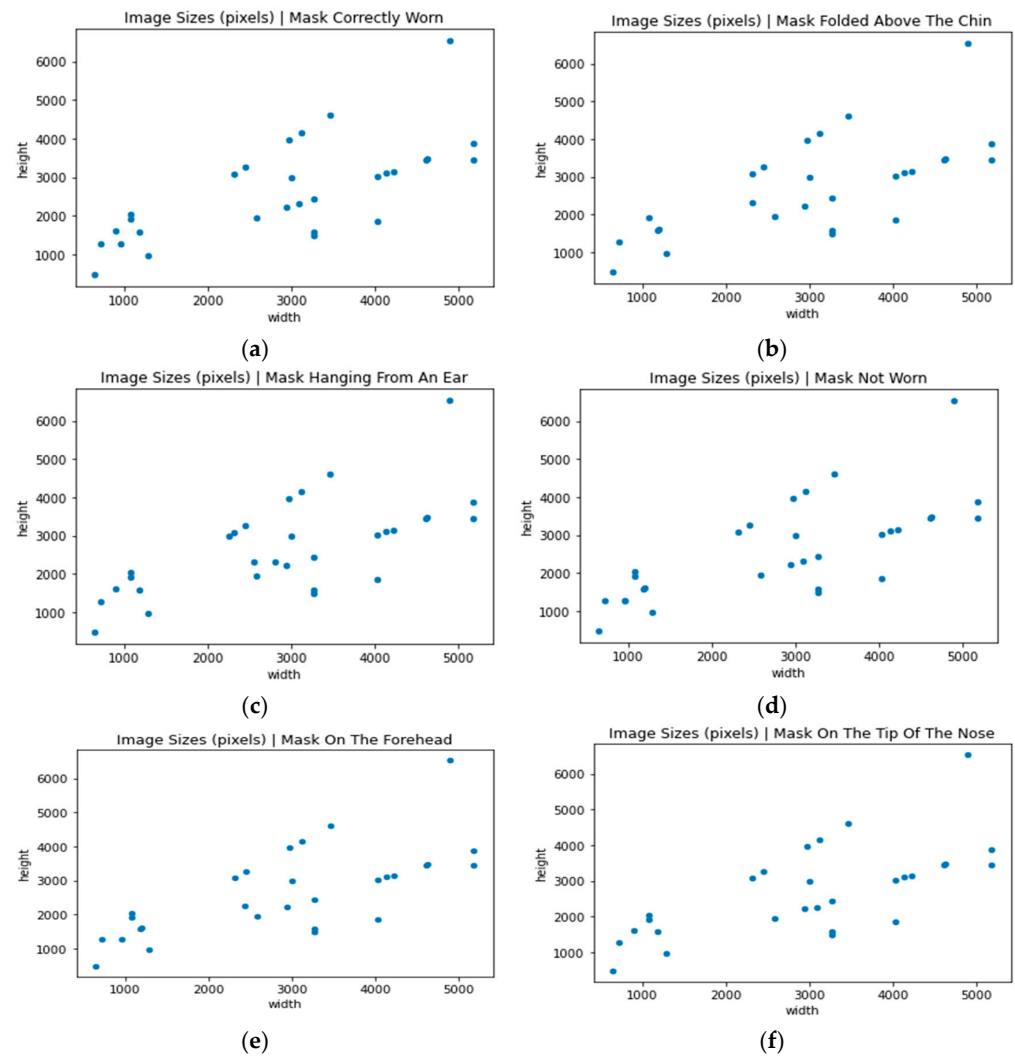


Figure 2. Cont.

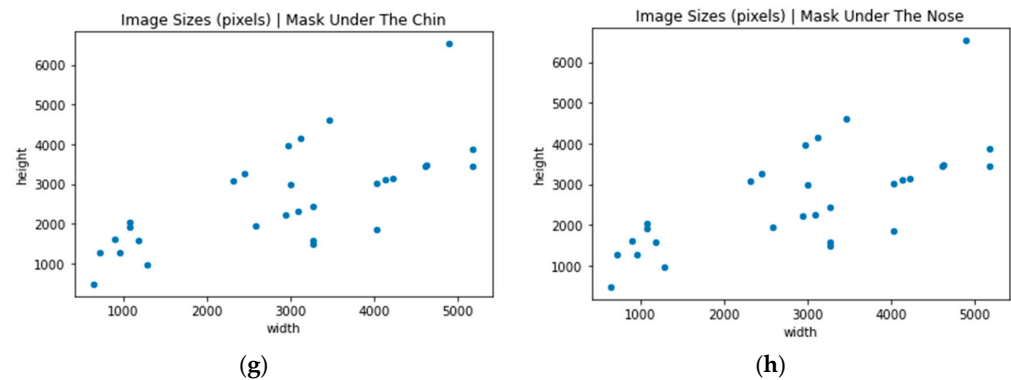


Figure 2. Scatter plots showing the resolution of images in each class.

3. Retain only the central region of the image;
4. Crop all the grayscale images using TensorFlow with a fraction of 0.5 (50%) to retain only the central region. In an image, the region of interest is the central region that occupies a person's face. Retaining only the region of interest helps the models to capture important patterns and eliminates the noise introduced by the irrelevant outer region.
5. Image resizing.
6. In the data understanding phase, we study found that most of the image resolutions were not the same and that their resolutions were very large huge. However, machine learning models can only receive inputs of the same size, and larger image resolution sizes require larger memory and processing capacity to process the images. Hence, image resizing was needed to scale down the images to the resolution recommended for image classification model training, which is 256×256 resolution.
7. Image flattening for use as model input.
8. The machine learning models require input in the form of a one-dimensional array. Hence, all the scaled-down images were flattened from two dimensions to one dimensions. A two-dimensional image with a resolution of 256×256 was resized to a one-dimensional image with a size of 65,536 columns. A single image was represented by a single one-dimension array with each column representing a single input feature. This is analogous to the use of a structured dataset where each row represents a single record and each column represents a single field of the record.
9. Input dimensionality reduction.
10. The input features that were obtained in this study for a single image were very large (65,536 columns) even after the images were cropped and scaled down. Input data with too many features are likely to overfit machine learning models during training by capturing noise or irrelevant information within the data, thereby causing the model to perform poorly on the testing data. To prevent this issue, Kernel Principal Components Analysis (KPCA) with Radial Basis Function dimensionality reduction technique, which is an extension of PCA that applies nonlinear transformation, was used to reduce the number of input features of all the images. This technique transforms large input features into a smaller number of principal components for use as new input features, where each principal component represents a percentage of the total variance captured from the data. All the input features values were first standardized to the range of 0–1 to ensure that features with higher pixel intensity would not dominate over features with smaller pixel intensity. The dimensionality reduction technique was then applied to the standardized input features, resulting in a total of 1220 principal components or features for each image.

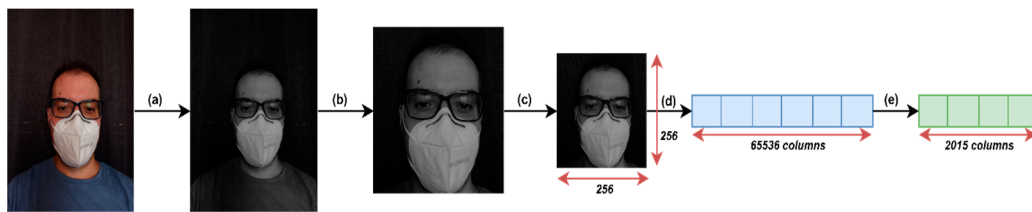


Figure 3. The image preprocessing steps.

3.3. Data Preparation

In this study, we were only interested in detecting people who wearing a mask correctly, not wearing a mask, or wearing a mask incorrectly. Thus, all the images containing 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 were merged into a single class called mask incorrectly worn, as all these images represent incorrect ways of mask-wearing. Merging these classes reduces the complexity of the resulting models. The total number of images in the mask correctly worn, mask not worn, and mask incorrectly worn classes after merging were 152, 113, and 957, respectively. As the sample number in the first two classes was lower than for the mask has incorrectly worn class, the image augmentation technique from the OpenCV library of horizontal and vertical image flipping was used to upsample the images in the two classes. After upsampling, the total number of images in the mask correctly worn, mask not worn, and mask incorrectly worn classes were 608, 452, and 957, respectively. K-fold cross-validation was used to divide the data into ten folds to ensure that different portions of the data were used for training and testing the model at different iterations.

3.4. Model Building

All the inputs features served as the independent variables to predict the three different classes: mask correctly worn (0), mask incorrectly worn (1), and mask not worn (2). From the literature review, five supervised machine learning models suitable for image classification were identified: Naïve Bayes, Support Vector Machine, Decision Tree, Random Forest, and K-Nearest Neighbors. All these models were built using the Scikit-Learn library in Python.

3.5. Model Evaluation

To assess the performance of the models, the evaluation metrics included accuracy, precision, recall, and F1-score. The scores obtained by each model were compared in order to find the best-performing one.

4. Results and Discussion

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

Table 1 shows the overall scores obtained by the Naïve Bayes, Support Vector Machine, Decision Tree, Random Forest, and K-Nearest Neighbors models. Table 1 clearly shows that the Decision Tree model outperforms the other four models in this image classification task. The training time of the K-Nearest Neighbors (KNN) is significantly lower than the rest of the models; however, it has the worst prediction performance, and the accuracy is lower than a random classifier. Although Support Vector Machine and Naïve Bayes performed better than KNN, their accuracy and precision are low.

Table 1. The overall performance scores obtained by the model.

Machine Learning Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Trained Time (s)
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 Neighbors (K = 5)	31.6	10.2	31.6	15.4	0.3

The tree-based algorithms, which are Decision Tree and Random Forest, performed better in this image classification task. Both algorithms achieved more than 80% in terms of accuracy, precision, recall, and F1-score, which is considered good model performance. Decision Tree had better results compared to Random Forest. For all evaluation metrics, Decision Tree achieved scores around 2% higher than Random Forest. Furthermore, the time taken to train a Decision Tree (50.79 s) was less compared to Random Forest (86.62 s). One possible reason that the performance of the Naïve Bayes, Support Vector Machine, and K-Nearest Neighbors models is worse than the tree-based algorithms is that our target classes were nonlinearly separable, resulting in better performance on the part of the nonlinear Decision Tree and Random Forest models.

Therefore, Decision Tree is the best model among these five models. The accuracy (%), precision (%), recall (%), F1-score (%), and training time (seconds) achieved by Decision Tree were 85.7, 85.9, 85.7, 85.7, and 50.79, respectively. The accuracy metric shows that, on average, 85.7% of the time the Decision Tree model is able to classify the image classes correctly. The precision score shows that, on average, 85.9% of the image's classes predicted by Decision Tree in fact have that class. In addition, it shows that false positives are low, as few images were wrongly classified. Next, the Decision Tree model'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 good performance in image classification.

When looking through the images where the Decision Tree model predicted the class wrongly, it was discovered that the model could predict "mask under the nose" images correctly as "mask incorrectly worn", while it wrongly predicted "mask correctly worn" images as "mask incorrectly worn" (refer to Figure 4). This might be due to the similar patterns between these two images, where the incorrect mask-wearing image on the left has the mask worn slightly below the nose and the mask correctly worn image on the right has the mask fully covering the nose. Thus, the model may not be able to distinguish the relevant patterns clearly.

**Figure 4.** Mask worn under the nose (left) and mask correctly worn (right).

To test whether the Decision Tree model can correctly predict unseen images, six mask-wearing images that consisted of mask correctly worn, mask incorrectly worn, and mask not worn were randomly selected from Google. After applying the same image preprocessing steps to the new images, the Decision Tree model was used to predict the images' target

class, and the model predicted all the images as “mask incorrectly worn”. This shows that the model is biased towards the “mask incorrectly worn” class; this situation might be caused by the insufficient number of training samples for the “mask correctly worn” and “mask not worn” classes, leading to the model coming up with the prediction “mask incorrectly worn” too frequently.

5. Conclusions

In this paper, we identified and implemented five machine learning algorithms for face mask wearing image classification tasks. Various preprocessing steps were applied to the dataset: color image to grayscale conversion, retaining the central region, image resizing, image flattening, and dimension reduction. Five machine learning algorithms (Support Vector Machine, K-Nearest Neighbors, Decision Tree, Random Forest, and Naïve Bayes) were used for face mask wearing image classification. From the results, the worst to the best models were K-Nearest Neighbors, Naïve Bayes, Support Vector Machine, Random Forest, and Decision Tree. The Decision Tree model was the best among all the models, achieving accuracy (%), precision (%), recall (%), F1-score (%), and training time (seconds) results of 85.7, 85.9, 85.7, 85.7, and 50.79, respectively.

When working on the implementation of models using Python, the study encountered processing capacity limitations using a personal laptop. The total size of the images was around 3.08 GB, despite having small samples (1222 images are analogous to 1222 records in a structured dataset), as most of the color images had high resolution. Processing these high-resolution images required significant processor power to speed up the image processing time. When using a laptop with 8GB RAM and a four-core processor with 2.4 GHz, reading the images into Python took up to 9 min to finish execution.

For future work, we propose increasing the number of samples in the “mask correctly worn” and “mask not worn” classes to ensure that the models have enough samples to learn to distinguish between the classes. However, in order to handle more images, more processing power is needed. To solve this issue, we propose utilizing cloud computing resources that can provide sufficient processing power to process large images faster without having to purchase a new more powerful computer. Lastly, in recent years many scientists have found that deep learning methods such as Convolutional Neural Networks (CNN) outperform machine learning techniques in image classification [6]. Hence, we propose experimenting with deep learning algorithms for face mask wearing image classification tasks in order to examine whether better performance can be achieved compared to the five machine learning algorithms used in this paper.

Author Contributions: Supervision, editing and revising draft, V.B.; Funding acquisition and revising: K.R.; Conceptualization, Methodology; Analysis; Drafting: H.Y.W., S.H.T., K.L.S. and W.K.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available by requesting the corresponding author vimala.balakrishnan@um.edu.my.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marceddu, A.; Ferrero, R.; Montrucchio, B. *Ways to Wear a Mask or a Respirator (WWMR-DB)* | *IEEE DataPort*; IEEE: New York, NY, USA, 2021. [[CrossRef](#)]
2. Carvalho, M.; Kaos, J. COVID-19: Wearing of Face Mask Remains Mandatory, Says Khairy | *The Star*. 24 March 2022. Available online: <https://www.thestar.com.my/news/nation/2022/03/24/covid-19-wearing-of-face-mask-remains-mandatory-says-khairy> (accessed on 14 September 2022).

3. Singh Chauhan, N. Naïve Bayes Algorithm: Everything You Need to Know-KDnuggets. Available online: <https://www.kdnuggets.com/2020/06/naive-bayes-algorithm-everything.html> (accessed on 14 September 2022).
4. Chugh, R.S.; Bhatia, V.; Khanna, K.; Bhatia, V. A Comparative Analysis of Classifiers for Image Classification. In Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 29–31 January 2020; IEEE: New York, NY, USA, 2020; pp. 248–253.
5. Goyal, H.; Sidana, K.; Singh, C.; Jain, A.; Jindal, S. A Real Time Face Mask Detection System Using Convolutional Neural Network. *Multimed. Tools Appl.* **2022**, *81*, 14999–15015. [[CrossRef](#)] [[PubMed](#)]
6. Huang, C.; Wang, Y.; Li, X.; Ren, L.; Zhao, J.; Hu, Y.; Zhang, L.; Fan, G.; Xu, J.; Gu, X. Clinical Features of Patients Infected with 2019 Novel Coronavirus in Wuhan, China. *Lancet* **2020**, *395*, 497–506. [[CrossRef](#)] [[PubMed](#)]
7. Jankovic, R. 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, 24 January 2019; pp. 119–127.
8. Kumar, S.; Khan, Z.; Jain, A. A Review of Content Based Image Classification Using Machine Learning Approach. *Int. J. Adv. Comput. Res.* **2012**, *2*, 55.
9. Naufal, M.F.; Kusuma, S.F.; Prayuska, Z.A.; Yoshua, A.A.; Lauwoto, Y.A.; Dinata, N.S.; Sugiarto, D. Comparative Analysis of Image Classification Algorithms for Face Mask Detection. *J. Inf. Syst. Eng. Bus. Intell.* **2021**, *7*, 56–66. [[CrossRef](#)]
10. OpenCV. Available online: <https://opencv.org/> (accessed on 23 May 2022).
11. Sabottke, C.F.; Spieler, B.M. The Effect of Image Resolution on Deep Learning in Radiography. *Radiol. Artif. Intell.* **2020**, *2*, e190015. [[CrossRef](#)] [[PubMed](#)]
12. Scikit-Learn. Available online: <https://scikit-learn.org/stable/> (accessed on 23 May 2022).
13. TensorFlow. Available online: <https://www.tensorflow.org/learn> (accessed on 23 May 2022).
14. Uddin, S.; Khan, A.; Hossain, M.E.; Moni, M.A. Comparing Different Supervised Machine Learning Algorithms for Disease Prediction. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 281. [[CrossRef](#)] [[PubMed](#)]
15. Utomo, M.N.Y.; Violita, F. Face Mask Wearing Detection Using Support Vector Machine (SVM). *Int. J. Inform. Dev.* **2021**, *10*, 72–81.
16. Vijitkunsawat, W.; Chantngarm, P. Study of the Performance of Machine Learning Algorithms for Face Mask Detection. In Proceedings of the 2020-5th International Conference on Information Technology (InCIT), Chonburi, Thailand, 21–22 October 2020; IEEE: New York, NY, USA, 2020; pp. 39–43.
17. World Health Organization. Weekly Epidemiological Update on COVID-19—22 March 2022. Available online: <https://www.who.int/publications/m/item/weekly-epidemiological-update-on-covid-19---22-march-2022> (accessed on 14 September 2022).
18. Wu, Y.-C.; Chen, C.-S.; Chan, Y.-J. The Outbreak of COVID-19: An Overview. *J. Chin. Med. Assoc.* **2020**, *83*, 217. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.