Mask Compliance Detection on Facial Images

Lorenzo Garbagna, Holly Burrows, Lakshmi Babu-Saheer, and Javad Zarrin

Anglia Ruskin University

Abstract. The Covid19 pandemic has significantly changed our ways of living. Government authorities around the world have come up with safety regulations to help reduce the spread of this deadly virus. Covering the mouth and nose using facial masks is identified as an effective step to suppress the transmission of the infected droplets from one human to the other. While the usage of facial masks has been a common practice in several Asian societies, this practice is fairly new to the rest of the world including modern western societies. Hence, it can be noticed that the facial masks are either worn incorrectly (or sometimes not worn) by a significant number of people.
Given the fact that the majority of the world population is only getting accustomed to this practice, it would be essential for surveillance systems to monitor if the general population is abiding by the regulatory standards of correctly wearing a facial mask. This paper uses deep learning algorithms to track and classify face masks. The research proposes a mask detection model based on Convolutional Neural Networks to discern between a correct usage of facial masks and its incorrect usages or even lack of it. Different architectures have been tested (even on real-time video streams) to obtain the best accuracy of 98.9% over four classes. These four classes include correctly worn, incorrectly worn on the chin, incorrectly worn on mouth and chin, and not wearing a mask at all. The novelty of this work is in the detection of the type of inaccuracy in wearing the face mask rather than just detecting the presence or absence of the same.

Keywords: Deep Learning, Computer Vision, Face Masks Detection

1 Introduction

The COVID-19 pandemic continues to challenge countries and governments to control the spread of the coronavirus. As one of the important control measures, the general public is advised to cover their mouth and nose using face masks to stop the spread of infectious droplets. The specified places where this is now a legal requirement may vary between nations. However, the general consensus in most of Europe is that face masks should be worn in smaller public spaces, any crowded outdoor settings, and all indoor public buildings [4].

A few nations around the globe are more accustomed to this practice; the use of facial masks in Chinese and Japanese populations can be traced back to the beginning of the 20th century [18], implemented as protection from seasonal flu

and as a societal ethical component. However, in countries where this practice is in its infancy, challenges arise where the masks are worn incorrectly, thus reducing their efficacy. This brings about further challenges of ensuring face masks are worn correctly where it is deemed necessary to effectively reduce the spread of the disease. Therefore, a solution is required to identify incorrect usage of masks when in public spaces. It would be inefficient, if not impossible for security professionals alone to monitor this especially among large crowds, or where this might pave way for possible confrontations. Considering this, technical solutions for automatically monitoring the correct usage of masks would be advantageous to reduce the spread of coronavirus. The goal of this study is to develop a system that is able to detect and classify if a person is wearing a mask correctly, or not. The system will also detect type of wrong usage if it is not worn correctly. The system is implemented using Convolutional Neural Networks (CNN) to identify the states in which a mask is found on a person's face. To demonstrate the application of this system in a real-world scenario, the model was also tested on live video stream to detect the changes in mask states in real-time. This automatic real-time monitoring system could be practically implemented in airports, supermarkets, workplaces and schools. The main novelty of this research lies in the ability to classify images into different categories of incorrectly worn masks, as opposed to a binary classification of mask being present or not.

2 Related Work

This section will look into the related research in the domain of face and face mask detection.

Loey et.al.[11] aimed to annotate medical face masks using real-life images. The research implements a transfer learning approach, developing a model comprising two parts. The initial phase employs feature extraction based upon ResNet-50, and the second detects the presence of a medical face mask, based on the state-of-the-art object detection system YOLO-v2. The dataset used a total of 1415 images, as an amalgamation of Medical Masks dataset (MMD) and Face Mask dataset (FMD), obtained from a Kaggle challenge. The size of the dataset is relatively small, mainly due to the scarcity of quality data in this new domain. This model was only able to achieve an accuracy of 81% and struggled to discern between surgical face masks and other types of masks. The Face Mask dataset from Kaggle was also utilised by Das et.al.[3] and this model was able to achieve a 94.8% accuracy on validation by using a cascade classifier to identify faces from the images and load them into the model. Another study was conducted by Loey et.al.[10], using feature extraction on a much larger dataset (11570 images) to feed different models and compare performances: decision trees, SVM and an ensemble method constituting k-NN, linear regression and logistic regression. This study was able to reach an accuracy of 99% on a simulated face mask dataset and on a real-world face mask dataset, with around 5000 images.

Comparatively, Mohan et.al.[12] built a CNN model that achieves 99.81% accuracy on a binary classification task, where the classes represent wearing a

mask, and not wearing a mask. The results obtained here outperformed the SqueezeNet model. The aim of this work entailed building a model that could be deployed at resource constrained endpoints. The final model had 128,193 trainable parameters and images were passed into the model at size of 32x32. Intricate data augmentation was used to increase the size of the dataset to 131,055 images. This work is demonstrative of CNN capabilities even in application areas where resource management is crucial to deployment.

Nagrath et.al.[13] proposed a deep learning model that makes use of Single Shot multibox Detector (SSD) for face detection in real-life images of peoples' faces with and without face masks. Transfer learning is applied, where the MobileNetV2 architecture is used for the classifier framework, alongside the pretrained weights from ImageNet. This model is suitable for real-time classification in the application domain due to its lightweight architecture. Due to the scarcity of suitably sized datasets in this area, the work involved using a combination of freely available datasets, such as those from Kaggle challenges. The authors of the work express their reluctance to use a dataset where masks are artificially added in the image. So 5521 images of real-life people wearing, and not wearing, masks were created. The work experimented with various pretrained models on the augmented dataset. Results showed that the proposed model outperforms LeNet-5 and AlexNet in accuracy, in addition to achieving the highest F1 score compared to LeNet-5, AlexNet, VGG16 and ResNet-50.

Chavda et.al.[2] present a dual stage CNN architecture for detecting facial masks. Firstly, a face detector identifies multiple faces in the same image as Regions of Interest (ROI). These are grouped and forwarded to stage 2 of the architecture, where the CNN classifies into a binary separation of masked or not masked. The output is the input images, where the faces are highlighted with a bounding box and their classification label. The CNN was trained with three popular classification architectures, namely DenseNet121, MobileNetV2 and NASNetMobile. The average inference speed of the three models was also measured, showing that DenseNet121 was the slowest at 0.353 seconds. It was concluded that NASNetMobile is the most suitable for applications operating in real-time.

Kayali et al.,[8] explore deep learning methods to accurately detect and classify face masks. To obtain a dataset suitable for the task, the researchers used images from the Labeled Faces in the Wild (LFW) database, and added face masks to the images of peoples' faces. Three classes were created: correct wearing, incorrect wearing, and no mask present (classes 0, 1, 2). Transfer Learning was utilised, whereby the performance of NASNetMobile and ResNet50 were compared. These pretrained models were chosen due to their contrast in depth of parameters; ResNet50 represents the performance of a deep network for this task, whereas NASNetMobile demonstrates lighter weight network potential. The images were sized at 128x128; models were trained for 200 epochs; and a small LR for Adam was used at 0.0000001. Interestingly, it took 80 minutes to train NASNetMobile, and just 60 minutes for ResNet50, even though the former is the lighter weight network. NASNetMobile showed poor performance for this

task, with accurately classifying just 33/499 for class 0 and 35/485 for class 1; but 100% accuracy for class 2. This concludes its' architecture is not suitable for this problem domain. On the other hand, ResNet50 demonstrated 92.38%, 89.48% and 93.61% for classes 0, 1, 2 respectively. The work concluded that this network has an overall classification accuracy of 92%.

Fasfous et al. [5] present a low-power Binary Neural-Network (BNN) to classify face-mask wearing, and the position of the mask on the face. Furthermore, the classifier was deployed to Edge Devices to mitigate chances of data exploitation, and maintain data privacy; the research also describes how using a BNN alongside this deployment method reduces the memory footprint of the network, as parameters are represented in the binary domain. The work used the MaskedFace-Net dataset, and reports that in its original form, a large class imbalance exists, where 51% of the dataset is dominated by correct wearing of face masks. To combat this, the larger classes were sampled randomly in order to increase the contribution of the smaller classes. Heavy data augmentation techniques were then applied to the now balanced data, resulting in 110k images for training and validation, with a large test set of 28k images. With the images sized at 32x32, the network was able to achieve a classification accuracy of $\tilde{9}8\%$ for the four mask-wearing positions on the face. The work boasts good model generalisability, and therefore reliability when presented with varying facial structures, hair types, skin tones, as well as age groups.

Bhuiyan et al. [1] develop an assistive system with Deep Learning which is used to classify the presence of face masks. The work employs a binary classification problem, where each face in an image receives a prediction of Mask or No Mask. The project involved extensive data analysis and preprocessing, where a web-scraping tool was used to pinpoint 650 images of people wearing, and not wearing, facial masks. The data was preprocessed to remove any images considered irrelevant to the task, resulting in 600 for training; there was an even distribution of 50% between the binary input classes. It was necessary to then label the acquired data to ensure its suitability to the task: the use of LabelIMG annotated all data samples. The authors describe the process of drawing bounding boxes in each image, where some contain multiple bounding boxes due to the need to identify any objects detected, and the presence of multiple people. With regard to model development and training, 4000 epochs of training facilitated by Google Colab achieved 96% accuracy and 0.073 loss. With this performance level, the research was able to progress to deploying the model to classify video captured in real-time, achieving on average 17 frames-per-second. Although the results show good promise, the authors conclude the paper with the fact that the dataset used to train is not highly varied. This is likely to be disadvantageous if the application is used in-the-wild, potentially meaning that people are entering crowded, or indoor spaces without wearing a face covering, which is detrimental to public health. Further work for this study is outlined as experimenting with varying object detection, such as RCNN, and YOLOv4 when available for public use.

Singh et al. [16] begin their works describing that current methods to address this problem domain mainly revolve around using simple CNN networks for binary classification of mask or no mask. However, the work advocates that the first step in the method should always be object detection, whereby bounding boxes are placed around faces in images. The classification of mask wearing should be the second step in the method, so that analysis of compliance can take place. This work brings its' focus to loss values to judge performance of each network; Transfer Learning with YOLOv3 achieved a validation loss of 0.25 comparatively to Faster RCNN validation loss of 0.15. The authors conclude that although the latter network has a better performance, for real-world deployment, YOLOv3 should be preferred due to its reduced inference time.

Koklu et al. [9] experiment with Transfer Learning, Long-Short Term Memory networks (LSTM), and bi-directionl LSTM networks for face mask determination. The work involved creating a dataset of 2000 images, where the same person is captured three times to create enough data for four classes: masked, non-masked, masked but with the nose exposed, and mask under the chin. A total of six experiments were carried out using two pretrained models: AlexNet and VGG16. The first approach was simple Transfer Learning, where the pretrained models are trained on the new dataset; the second involved removing the classification header for both pretrained models and replacing it with LSTM structure; the third, replacing the classification layers with bi-directional LSTM architecture. All experimental results achieved accuracy scores north of 90%, with the most modest result coming from transfer learning with AlexNet at 90.33%, and the best, 95.67% with VGG16 using bi-directional LSTM as the classification layer. The best recall was for no mask present, and the worst was for mask under the chin.

3 Data

3.1 Description

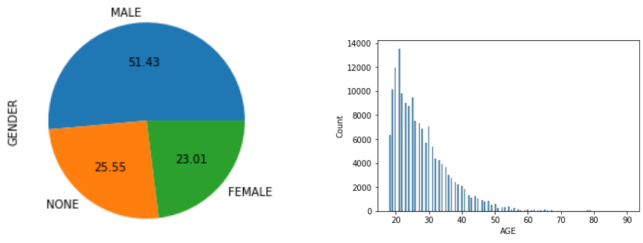
The dataset used, at the time of writing, is the largest available containing images of people wearing face masks in real life. There are 250,000 images available in total, comprising 28,000 different people, alongside showing four varying types of face mask. The data is distributed between seven separate folders, available to download from Kaggle [15]. The data is spread across four classes: 0- No mask, 1- Mask but nose and mouth exposed, 2- Mask but nose exposed, 3- Mask is worn correctly. See figure 1 for examples of each class.

The research is limited by resource constraints, thus unable to utilise the available 250,000 images, and so makes use of parts 1-4 of the available data. This totals $\tilde{1}$60k images. 30,000 of these images were reserved for the final testing dataset. Allowing 20% of the training dataset for validation resulted in $\tilde{1}$04k for training. The original images were of varying sizes, and very large, most exceeding 1024 for height and width. Inspection of the dataset demonstrated an even distribution between the four classes, with $\tilde{3}$2,438 images per class. This will be advantageous to the performance of each model; in work by [7] it is explained

Fig. 1. Kaggle Dataset Training Examples

that for imbalanced datasets, networks have a tendency to over-classify samples consistently to the class with the most samples. In such circumstances, the minority class is frequently classified incorrectly, resulting in poor performance on unseen data. However, the class balance shown in this dataset mitigates the chance of this occurrence.

For neural networks to show the best performance when deployed as an application, good generalisation to unseen data is imperative. This can be improved when the training data has a large variation of samples. A variable in this dataset is the gender of the person shown in each image; men and women present different facial characteristics, thus providing the classifier with some variation. The majority of images in this dataset were labelled with the gender of the person, however 25.55% were marked as None. The dataset is heavily dominated by images of males at 51.43%, and only 23.01% are of women. Refer to figure 2(a) for the distribution. An additional variable observed within the dataset is the age of the person in the image. Figure 2(b) shows the distribution of age groups in the training data. It shows that images of people aged 20-30 years old dominate the data, but the range spans 18-79 years. On initial inspection, analysis showed that some images contained incorrect values for age, such as 2020. This inaccurately skewed the analysis, so a Python script was used to find images where the age value exceeded 100, and the persons age was simply estimated.



(a) Gender Distribution (b) Age Distribution

Fig. 2. Data Analytics

3.2 MaskedFace-Net

MaskedFace-Net is a dataset comprising 137,016 images of people's faces, all of which have had a surgical mask photoshopped onto them. There are 4 possible classes at this stage: (1) the person in the image is wearing the face mask correctly, with chin, mouth and nose covered; (2) the mask covers the chin only; (3) the mask covers the mouth and chin only; (4) the mask covers the nose and mouth, leaving the chin exposed. The correctly masked class dominates this dataset at 49%. Figure 3 shows some sample images.

Fig. 3. MaskedFace-Net Image Samples

3.3 Preprocessing

This section describes the preprocessing applied to the datasets before model development and training could commence. Firstly, all images were resized to a uniform 300x300; this value was chosen so that significant experimentation could be carried out with respect to imposed hardware constraints, whilst maintaining the significant features of the data. Second, for ease of implementation, the images were organised into folders corresponding to their class. This was achieved through creating a script that extracted the class label from the filename, and using the `os` library to iterate files and move to a specified directory.

4 Model Experimentation

The work experiments with varied implementations of Convolutional Neural Networks to classify input images into one of four classes. The model demonstrating the best performance during testing is used to classify input captured from real-time video. Each model and it's performance in relation to accuracy and loss on the training and validation set are explained and analysed.

4.1 Training and Validation

Three CNN models were trained to test for the highest accuracy.

The total number of images used for training is ~130k, 20% of which are used in the validation set. Graphs plotting the train accuracy against the validation accuracy are recorded, along with the loss. All models consisted of 2D-Convolutional layers, doubling the number of filters at every layer: the activation function used is ReLU and padding has been set to 'same'. The padding setting is set this way to enable the application for the video-stream mask detection to work correctly, as difference in padding would result in the methods implemented with OpenCV having inaccurate image shapes sent to the model for classification. MaxPooling2D with a size of 2x2 was implemented after every convolutional layer. Inputs are flattened after the filters have been applied, and the data is passed into a Dense layer before classification, a Dense layer with 5 neurons using the SoftMax activation.

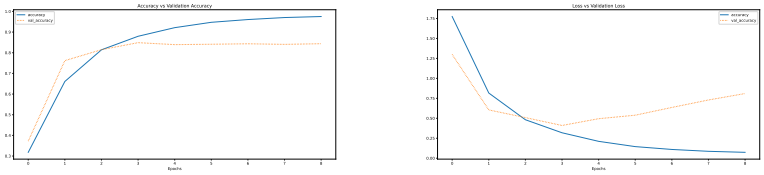
4.2 Model A

The first model used input images in the grey-scale colour space and had a total of 2,827,205 trainable parameters. The model architecture is shown in table 1

Layer 1: Convolutional	Conv2D(filters=16)
Layer 2: Pooling	MaxPool2D(2x2)
Layer 3: Convolutional	Conv2D(filters=32)
Layer 4: Pooling	MaxPool2D(2x2)
Layer 6: Convolutional	Conv2D(filters=64)
Layer 7: Pooling	MaxPool2D(2x2)
Layer 8: Flatten	
Layer 9: Dense	Neurons=32
Layer 10: Dense (SoftMax)	Neurons=4

Table 1. Model-A Layers

Figure 4 shows the training accuracy and loss against validation accuracy and loss respectively. The accuracy of the model on the train data grows over the specified number of epochs. The accuracy of the validation set is steadier but averagely lower, which could indicate that the model is over-fitting slightly. The training loss decreases steadily over time while the validation loss decreases only for three epochs, after that it increases constantly without improving.



(a) Training vs Validation Accuracy (b) Training vs Validation Loss

Fig. 4. Model-A Training

4.3 Model B

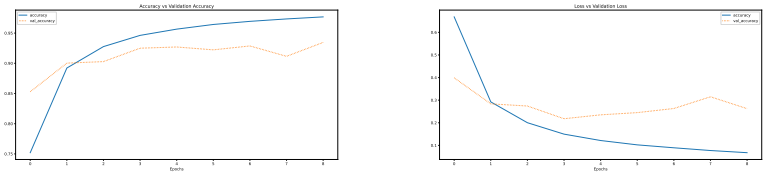
The second model also used input images in the greyscale colour-space and had a total of 1,424,453 trainable parameters. The model architecture is shown in table 2.

Layer 1: Convolutional	Conv2D(filters=16)
Layer 2: Pooling	MaxPool2D(2x2)
Layer 3: Convolutional	Conv2D(filters=32)
Layer 4: Pooling	MaxPool2D(2x2)
Layer 6: Convolutional	Conv2D(filters=64)
Layer 7: Pooling	MaxPool2D(2x2)
Layer 8: Convolutional	Conv2D(filters=128)
Layer 9: Pooling	MaxPool2D(2x2)
Layer 10: Flatten	
Layer 11: Dense	Neurons=32
Layer 12: Dense (SoftMax)	Neurons=4

Table 2. Model-B Layers

Figure 5 shows the training accuracy against validation accuracy and training loss against validation loss respectively. There is an improvement in this model given the fact it is over-fitting less than model-A. As shown in Model-A, the validation loss in this model stops decreasing after epoch 3, but it reaches a

lower value and it increases with a smaller magnitude compared to Model-A.



(a) Training vs Validation Accuracy (b) Training vs Validation Loss

Fig. 5. Model-B Training

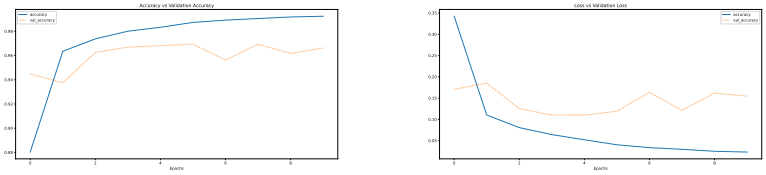
4.4 Model C

Table 3 shows the architecture for the model-C with RGB input images and the total number of 3,047,589 trainable parameters. Note that Batch Normalisation was implemented after each Convolutional layer in the network.

Layer 1: Convolutional	Conv2D(filters=16)
Layer 2: Pooling	MaxPool2D(2x2)
Layer 3: Convolutional	Conv2D(filters=32)
Layer 4: Pooling	MaxPool2D(2x2)
Layer 6: Convolutional	Conv2D(filters=64)
Layer 7: Pooling	MaxPool2D(2x2)
Layer 8: Convolutional	Conv2D(filters=128)
Layer 9: Pooling	MaxPool2D(2x2)
Layer 10: Convolutional	Conv2D(filters=256)
Layer 11: Pooling	MaxPool2D(2x2)
Layer 12: Flatten	
Layer 13: Dense	Neurons=128
Layer 14: Dense (SoftMax)	Neurons=4

Table 3. Model-C Layers

Figure 6 shows the training accuracy against validation accuracy and training loss against validation loss respectively. The performances are superior compared to the previous two models, and both accuracy and loss values for training and validation present a smaller gap then previous architectures.



(a) Training vs Validation Accuracy (b) Training vs Validation Loss

Fig. 6. Model-C Training

This model showed good performance during validation; 0.9681 and 0.1101 for validation accuracy and loss respectively. The model was saved as a JSON file along with the weights.

MobileNetV2 and ResNet50 have also been trained using Transfer Learning. Table 4 compares the performance of all five experimental methods.

	Model-A	**Model-B**	**Model-C**	**MobileNet-V2**	**ResNet50**
Color Mode	Greyscale	Greyscale	RGB	RGB	RGB
Total params	2,827,172	1,424,420	3,049,444	10,294,788	24,406,916
Time/Epoch	187s	160s	277s	171s	750s
Train Accuracy	0.8797	0.9462	0.9833	0.7433	0.9502
Validation Accuracy	0.8488	0.9250	0.9681	0.7277	0.9261
Train Loss	0.3188	0.1498	0.0521	0.6514	1.924
Validation Loss	0.4101	0.2179	0.1101	0.6855	4.0052

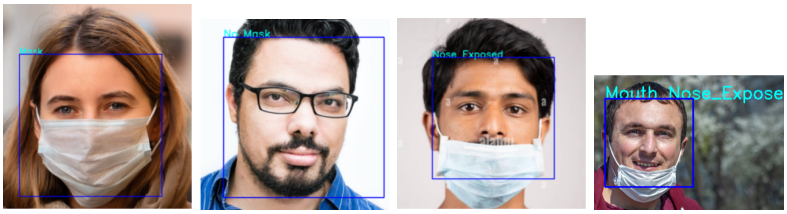
Table 4. Model Comparison

5 Application

This section describes the use of the proposed end application as a proof of concept. A script takes the JSON and weights files of each model, and loads a prediction method that returns the state of the mask. The argmax function from the Numpy library (`np.argmax`) is used to load the ID of the class (0 to 3), instead of the probability for each class. OpenCV [14] is then used to import the Haar classifier [17]: when the video-stream from the webcam is activated, or an image file is presented, the classifier detects any faces present. The Region of Interest (ROI), the face inside the bounding-box, is resized to 300x300 and fed to the prediction method; the state is classified, and the text associated with the ID of the class is shown on top of the bounding box around the face. The models

were tested on unseen images and webcam footage and results are described in a latter section of the paper. Deployment of the classifier on unseen images and webcam feed is representational of how it might behave in-the-wild.

Figure 7 shows an example of the concept used for the application by using 4 images with 4 different mask states.



(a) Mask (b) No Mask (c) Nose (d) Mouth Nose

Fig. 7. Example of Application by Image Classification

6 Results

Table 5 shows the result of the CNN models using the unseen test set: 7,500 images per class. Between the three custom models, the only one using RGB images as input, Model-C, reaches the highest accuracy score of 0.9663 and lowest loss value of 0.1272. Comparing the two models that used greyscale images, Model-B outperformed Model-A by reaching an accuracy of 0.9241 against 0.8488: even with fewer parameters, the additional Convolutional Layer allowed the network to learn more significant features, thus generalising better.

	Color Mode	**Accuracy**	**Loss**
Model-A	Greyscale	0.8488	0.4104
Model-B	Greyscale	0.9241	0.2323
Model-C	RGB	0.9663	0.1272
MobileNet-V2	RGB	0.7353	0.6772
ResNet50	RGB	0.9261	4.2984

Table 5. CNN Models Test Results

Due to previous publications using pretrained models on new datasets, this research also implemented two of these architectures: MobileNetV2 and ResNet50. The first one under-performed, positioning itself last: the smaller image sizes (224x224) and fewer parameters compared to other models influenced its poor accuracy score of 0.7353. On the other hand, ResNet50 was the second best

model for accuracy at 0.9261, but presented a significant loss of 4.2984: the model could predict the class correctly most of the time, but it displayed high uncertainty about the decision. To get a better understanding of these results, confusion matrices for each architecture have been plotted in Figure 8 and their numerical values noted in tables 6-10.

(a) Model-A (b) Model-B (c) Model-C (d) MobileNet (e) ResNet50

Fig. 8. Confusion Matrices

	Mask	*No Mask*	*Mouth Nose*	*Nose*
Mask	6701	184	61	554
No Mask	105	6228	955	212
Mouth Nose	99	979	6100	322
Nose	609	285	171	6435

Table 6. Model-A Confusion Matrix

	Mask	*No Mask*	*Mouth Nose*	*Nose*
Mask	7167	91	18	224
No Mask	40	7077	325	58
Mouth Nose	44	1060	6286	110
Nose	135	103	69	7193

Table 7. Model-B Confusion Matrix

	Mask	*No Mask*	*Mouth Nose*	*Nose*
Mask	7367	34	4	95
No Mask	23	7320	133	24
Mouth Nose	18	500	6906	76
Nose	69	23	13	7395

Table 8. Model-C Confusion Matrix

	Mask	*No Mask*	*Mouth Nose*	*Nose*
Mask	6057	396	219	828
No Mask	575	5329	1325	271
Mouth Nose	363	920	5796	421
Nose	1555	454	613	4878

Table 9. MobileNet Confusion Matrix

	Mask	*No Mask*	*Mouth Nose*	*Nose*
Mask	6900	10	13	577
No Mask	29	7007	414	50
Mouth Nose	18	496	6777	210
Nose	349	13	38	7100

Table 10. ResNet50 Confusion Matrix

Tables 6-10 confirm what is shown by the evaluation on the test sets: Model-C has very high performance in terms of any confusion. The model can identify nearly perfectly the Mask and Nose classes by classifying correctly 7367 and 7395 respectively, against the 7500 total images per class. Although it misclassified

180 images for the No Mask class and 594 for the Mouth Nose class, the generalisation on test and unseen images was highly acceptable. The only other model that has acceptable performances is ResNet50: although performing worse, especially in the Mouth Nose class where it misclassified 723 images, it confirms the high accuracy scored on the dataset, even considering its low confidence in classification. The other models present a lot of confusion between the classes, classifying incorrectly many of the images, particularly in the Mouth Nose class. This is most likely due to the additional features needed to thoroughly map the contours on the masks, and to detect the edges on the chin.

On the other hand, Table 11 shows the result of the three same models tested on the MaskedFace-Net test set: at first glance Model-C achieves an even higher accuracy and a lower loss compared to the second dataset, scoring an accuracy value of 0.9896 against 0.9706 and a lower loss at 0.0469 against 0.1115. Model-A and Model-B also showed better performances on this dataset: the lowest accuracy scored was achieved by Model-A at 0.9777. Although more accurate, the MaskedFace-Net models performances on generalisation were poor, as described in the Discussion.

	Color Mode	**Accuracy**	**Loss**
Model-A	Greyscale	0.9777	0.1239
Model-B	Greyscale	0.9833	0.0892
Model-C	RGB	0.9896	0.0469

Table 11. CNN Models Test Results for MaskedFace-Net

6.1 Real-Time Classifier

The web framework Flask [6] was used to show the feed captured by the webcam to the internet browser, so the output can be shown to the user. The output consists of a bounding box around the face(s), and the class prediction from the model in real-time, as demonstrated in Figure 9.

(a) Mask (b) Chin (c) Mouth Chin

Fig. 9. Real-Time Mask Classification Examples

Although working correctly, some limitations have been observed. Utilising the model trained on MaskedFace-Net, for the video-stream, a greyscale model trained on four classes, without the no_mask label, have been utilised. The Haar classifier from OpenCV used here had some constraints. It has been noted that in certain conditions of poor or excessive lighting, the classifier had difficulties to detect both the face and the mask of the subject.

7 Discussion

This work contributes a novel perspective for the application domain of Face Mask Classification. It uses a large dataset consisting of images where people are wearing physical face masks, in comparison to other relevant work that uses a photoshopped technique. This work is thus more representational of how a system such as the proposed would behave if deployed in a real-life setting, such as entrances to public buildings and transport. The paper provides an in depth analysis of various CNN performance in the application domain. Custom architectures are presented for the task; the performance of varying pretrained models with transfer learning are investigated to understand the capabilities of shallow and deeper networks; and a comparison between available datasets for the task is provided. The best performing network, Model-C, was trained using RGB images to increase the numbers of trainable parameters in order to improve training. By passing RGB inputs, time per epoch increases slightly but the architecture is able to learn more features by mapping the colours between the persons' skin and the mask. Another advantage is the possibility for the network to learn different mask colours in relation to skin tones that might closely resemble the mask, whereas a greyscale model might get confused by the closer intensity of the pixel values. Although the model trained on MaskedFace-Net outperformed its results using the second dataset, some major drawbacks have been found. Due to the nature of MaskedFace-Net, during different tests it has been observed that the model often struggles to correctly identify images where the person has a darker skin tone, especially if combined with a darker face mask. Furthermore, as the images contained only surgical masks applied with Photoshop, the model struggles to generalise to other types of masks. Due to these problems, Model-C generalises best when trained on the real masks dataset, as it is able to achieve a high accuracy score whilst also being able to categorise different types of masks applied to various ethnicities. This area of research is considered relatively new given the short amount of passed time since the beginning of the pandemic. Therefore, the available data varies in size and suitability. The MaskedFace-Net dataset provides a large pool of images, and, to the best of the authors knowledge, was the first to cover all variations of incorrect mask wearing behaviour. However, due to the photoshopped masks, it lacks the degree of realism that the main dataset used in this paper provides. In addition, MaskedFace-Net also demonstrates a large imbalance between classes, where 51% represents incorrectly masked faces, but only 10% of this is populated by nose exposure images. Without addressing the issue, this will inevitably lead

to under-performing models. Using the other dataset somewhat mitigates this issue, proven through an improved generalisability.

8 Conclusion and Further Work

Multiple neural networks were trained using a large dataset consisting of people wearing real masks in one of four ways, to detect the variations in mask-wearing behaviour. The experimentation makes use of three models using a simple CNN architecture, where two of these networks use greyscale images, and the remaining using RGB input. The models were compared and their performance evaluated against existing pretrained model performance with transfer learning, namely MobileNet and ResNet50. These specific networks were chosen to compare and contrast the abilities of shallow networks, and those using more layers, for this specific problem domain. The work concludes that a simpler CNN architecture taking RGB images as input yields the best performance. This particular model was able to classify images in the test dataset with an accuracy of 96.63%. The OpenCV implementation demonstrated that the systems' capability to classify mask state accurately in real-time, promoting the proof of concept. Taking the limitations of both the model and application into consideration, further improvements can be applied to both. The dataset can be expanded by introducing more pictures with different kinds of masks, apart from surgical ones. Data augmentation could be applied in case the number of new pictures would not be enough to bring any significant improvements to the model. On the application side, a new face detection system could be applied to reduce the limitations imposed by the Haar classifier. The work could benefit from testing the performance of other pretrained models, such as NASNetLarge, providing that the suitable resources are available for such heavy computation.

References

[1] R. Bhuiyan, S A. Khushbu, and S. Islam. "A Deep Learning Based Assistive System to Classify COVID-19 Face Mask for Human Safety with YOLOv3". In: *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. 2020, pp. 1–5. DOI: 10.1109/ICCCNT49239.2020.9225384.

[2] A. Chavda et al. "Multi-Stage CNN Architecture for Face Mask Detection". In: *2021 IEEE 6th International Conference for Convergence in Technology (I2CT)*. Pune, India, 2021, pp. 1–8.

[3] A Das., M. Ansari, and R. Basak. "Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV". In: *2020 IEEE 17th India Council International Conference (INDICON)*. New Delhi, India, 2020, pp. 1–5.

[4] European Centre for Disease Prevention and Control. *Using face masks in the community: first update*. 2021. URL: https://www.ecdc.europa.eu/sites/default/files/documents/covid-19-face-masks-community-first-update.pdf.

[5] N. Fasfous et al. "BinaryCoP: Binary Neural Network-based COVID-19 Face-Mask Wear and Positioning Predictor on Edge Devices". In: *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. 2021, pp. 108–115. DOI: 10.1109/IPDPSW52791.2021.00024.

[6] *Flask*. 2010. URL: https://flask.palletsprojects.com/en/2.0.x/.

[7] J M. Johnson and T M. Khoshgoftaar. "Survey on deep learning with class imbalance". In: *J Big Data* 6.27 (2019), pp. 1–54. DOI: 10.1186/s40537-019-0192-5.

[8] D. Kayali, K. Dimililer, and B. Sekeroglu. "Face Mask Detection and Classification for COVID-19 using Deep Learning". In: *2021 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. 2021, pp. 1–6. DOI: 10.1109/INISTA52262.2021.9548642.

[9] M. Koklu, I. Cinar, and Y S. Taspinar. "CNN-based bi-directional and directional long-short term memory network for determination of face mask". In: *Biomedical Signal Processing and Control* 71 (2022), p. 103216.

[10] M. Loey et al. "A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic". In: *Measurement* 167 (2021), p. 108288.

[11] M. Loey et al. "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection". In: *Sustainable Cities and Society* 65 (2021), p. 102600.

[12] P. Mohan, A J. Paul, and A. Chirania. *A Tiny CNN Architecture for Medical Face Mask Detection for Resource-Constrained Endpoints*. 2020. URL: https://arxiv.org/abs/2011.14858.

[13] P. Nagrath et al. "A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2". In: *Sustainable Cities and Society* 66 (2021), p. 102692.

[14] *OpenCV*. 2021. URL: https://opencv.org/.

[15] K. Roman. *500 GB of images with people wearing masks. Part 1*. 2021.

[16] S. Singh et al. "Face mask detection using YOLOv3 and faster R-CNN models: COVID-19 environment". In: *Multimedia Tools and Applications* 80 (2021), pp. 19753–19768.

[17] P. Viola and M. Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001*. 2001, pp. 511–518.

[18] J. Yang. *A quick history of why Asians wear surgical masks in public*. 2014.