Data Mining, Machine Learning, and Deep Learning

Face Mask Detection

Exam – Written Product

Students & Student Number: Alexander Van Le (119220), Frederik Gaasdal Jensen (141628) & Jacob Leopold Hinrichsen (120499)

Education: MSc. in Business Administration and Data Science

Date of Submission: 26-05-2021

Number of Characters: 33,111 (14.9 standard pages)

Number of Pages: 12

Face Mask Detection

Authors
Alexander Van Le (119220),
Frederik Gaasdal Jensen (141628),
Jacob Leopold Hinrichsen (120499)
{alle17ad, frje20ah, jahi17ab}@student.cbs.dk

Students - Copenhagen Business School
MSc. Business Administration and Data Science

Abstract

The Covid-19 pandemic has increased the importance of wearing a face mask to minimise the spread of the disease. In this paper, we are testing the robustness of Convolutional Neural Network (CNN) models in classifying images of people wearing a face mask or not. We deploy four different models: a CNN, a pre-trained CNN, a pre-trained ImageNet-InceptionV3, and an SVM model. We train the models using multiple datasets (n $\approx$ 40,000) and deploy standard image normalisation and augmentation techniques. We find that the InceptionV3 model achieved the best performance with an F_1 score of 0.993. Finally, to deploy such a model on a national scale, we see that the model does not need to be more complex than the InceptionV3 model. But we recommend that when training such a model, the dataset should be corrected for biases in terms of sex, skin colour, race and so on, as we see that the models tend to become biased towards these features if they are not balanced.

Keywords: Face Mask Detection, Convolutional Neural Network, Image Classifcation, Data Augmentation, Feature Analysis, ImageNet, InceptionV3

1 Introduction

The outbreak of Covid-19 has caused a worldwide pandemic affecting everyone. Over 3.4 million people have passed away due to the respiratory infectious disease, and this number grows every day. Most countries have found it necessary to implement emergency sanitary rules. One of the most common tools to prevent the spread of Covid-19 has been the face mask. Research has shown that this initiative is an effective tool in reducing the spread of viruses like Covid-19 (Cowling et al., 2009; Tracht et al., 2010). Using face masks is not a novel concept. Some jobs have been using face masks for a while, like doctors and dentists. There exist many jobs that employ strict sanitation rules, thereby requiring the use of face masks.

This paper tests the robustness of a Convolutional Neural Network in identifying whether a person is wearing a mask or not, which can have several uses cases. First, it can help scientists calculate the number of people wearing masks in certain areas. Furthermore, it can be used in industries with strict sanitation rules. Some factories use industrial machines that scan a user's face and cannot be started without identifying a mask on the user. In this paper, four algorithms are trained using around 40,000 images. Firstly, we deploy two custom CNN models, one with transfer learning and one without. Afterwards, we benchmark these models against an SVM model and a pre-trained InceptionV3 model.

2 Related Work

Multiple new publications focus on identifying whether people wear a face mask. The common goal of all related work within the subject is to decrease the spread and transmission of Covid-19. Chowdary et al. (2020) have developed an algorithm that detects if a person is wearing a mask or not. The algorithm uses the pre-trained convolutional neural network InceptionV3 with five added layers to classify the images. Their dataset is a simulated masked face dataset (SMFD). They increase the size of their dataset by introducing image augmentation such as rotation and zooming. Loey et al. (2021) also applied a

pre-trained network, in this case, ResNet-50. They remove the network's last layer and introduce machine learning methods such as a support vector machine and decision tree. The models are trained using two datasets: an SMFD and a real-world masked face dataset (RMFD). Loey et al. (2021) achieved a test accuracy of 99.76% using ResNet-50 and an ensemble classifier on an RMFD.

Suresh et al. (2021) recognises the practical implications of the issue. They investigate how face mask identification can be implemented to run real-time on CCTV notifying relevant authorities when people are not wearing masks. To accomplish this, they implement a lightweight pre-trained CNN called MobileNetV2 and further trains it with SMFD.

Qin and Li (2020) expands on the issue. They recognise that wearing a mask incorrectly is as good as not wearing a mask. Therefore, they introduce a new class for people not wearing the mask correctly. Before training and testing their network, they transform their images into super-resolution using a CNN with convolutional layers as autoencoders and deconvolutional layers for up-sampling. For image classification, they apply the MobileNetV2 lightweight CNN. The model is trained using transfer learning, training the model with images with no mask for the network to gain face-detection capabilities and reduce overfitting. Qin and Li (2020) found that MobileNetV2 contains a small fraction of the parameters that InceptionV3 contains, resulting in a running time for single image classification to be half of the Inception-V3. Furthermore, they conclude that using super-resolution before classification causes the CNN to achieve higher accuracy.

3 Conceptual Framework

3.1 Data Preparation

The data preparation consists of three main parts: data filtering, data normalisation, and data augmentation. See Figure 1. Our data points consist of images displaying people with and without face masks. We need to ensure that the images are of good enough quality before using them to train the classification models. We use different techniques in order to filter out bad images. After filtering the data, we need to make sure that the data is normalised so that the rest of the pipeline can work with it no matter the source of the data. Finally, before training the model, we need to ensure that we have enough images in terms of numbers and variety to ensure that the models can generalise as much as possible. This is achieved using data augmentation.

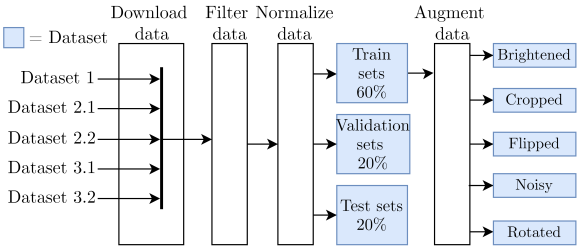


Figure 1: Data Preparation Process

3.2 Training Strategy

This paper will train various models, three of which are based on Convolutional Neural Networks and one of which is based on Support Vector Machine. All of these models will use the same training set, except for one of the CNN models, which we will pre-train using another data set to which the other models do not have access. See Figure 2.

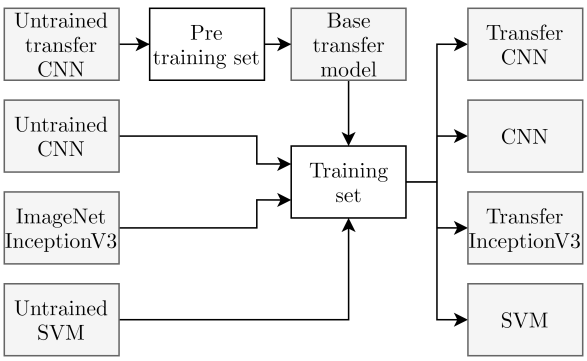


Figure 2: Training Strategy

4 Methodology

4.1 Dataset Description

The algorithms used in this article have been trained and tested using several datasets. In total, five different public datasets have been applied. Each dataset varies from the others in different ways. To gain consistency throughout the article, each dataset has been given a unique ID by which the dataset will be referred to. Table 1 displays the IDs of the five datasets, the number of images, and the nature of the images.

ID	Images	Type
Dataset 1	20,000	SMFD
Dataset 2.1	11,792	RMFD
Dataset 2.2	1,006	RMFD
Dataset 3.1	6,024	RMFD
Dataset 3.2	853	RMFD

Table 1: An Overview of the Datasets

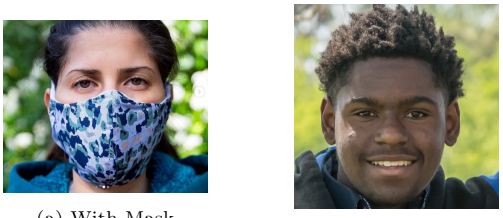
Dataset 1 contains high-resolution images of people with augmented masks that a Generative Adversarial Network has produced. All images have a width and height of 1,024, making them square. Figure 3 displays a sample of a picture from each category in the dataset. Observe how figure (a) is displayed with a generated mask.

(a) With Mask (b) Without Mask

Figure 3: Sample from Dataset 1

Dataset 2.2 contains images that have been handpicked from various image sources and manually annotated. Masked images contain real masked people. Dataset 2.1 contains similar images as 2.2. Figure 4 presents a sample from each class.

There are some differences in the image quality between the classes. Figure 5 presents



(a) With Mask (b) Without Mask

Figure 4: Sample from Dataset 2.2

a scatter plot showing the aspect ratio[1] and size. What can be observed in the plot is that images without a mask are generally smaller than images with masks. Additionally, there is a large variance in the sizes of images labelled with a mask. We see that there is bias in the data in terms of the image size. Smaller images tend to be the images without masks, and therefore the model might pick up this unfortunate trend. We take no further actions against this bias, but we are aware that it might skew the model's predictions.

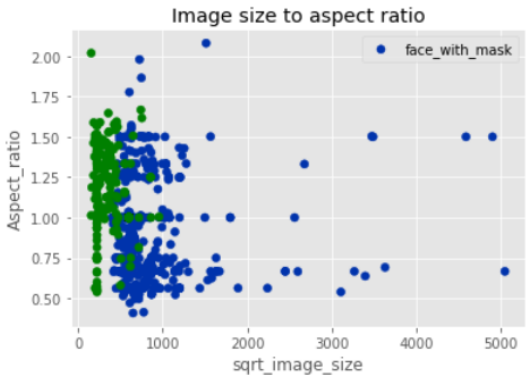


Figure 5: Size and Aspect ratio - Dataset 2.2

Dataset 3.1 differs from the datasets mentioned above by having multiple faces in the images. The images are accompanied by metadata detailing the pixel coordinates for each face in the dataset and whether the person is wearing a mask. It should be noted that some faces are tiny in terms of pixels, which will be problematic when using it to train the model. This problem will be expanded in the Data Filtering section. Figure 6 presents a sample from each class. Dataset 3.2 contains similar images with multiple faces.

[1]Height divided by the width.

(a) Image 1

(b) Image 2

Figure 6: Sample from Dataset 3.1

4.2 Data Preprocessing

4.2.1 Data Filtering

Before describing the filtering process, the image cropping in dataset 3.1 and 3.2 must be discussed. Since each image can contain multiple faces, we generate a new image for each face. This is done by using the position of each face in each image given in the metadata. This increases the size of both datasets. Table 2 shows the total amount of faces and the distribution between mask-wearing and non-mask wearing images.

ID	Faces		
	Total	With mask	Without mask
Dataset 1.1	20,000	10,000	10,000
Dataset 2.1	11,612	5,703	5,909
Dataset 2.2	1,006	503	503
Dataset 3.1	5,688	4,135	1,553
Dataset 3.2	2,259	1,926	333
Total	40,565	22,267	18,298
% Total	100%	54.9%	45.1 %

Table 2: Label Split

It can be noted that the data is somewhat unbalanced. There are slightly more images of faces with masks compared to images of faces without masks. We will not take further actions to ensure that the dataset is more balanced, but we are aware of issues related to unbalanced datasets. This is further discussed in the Results section.

To ensure optimal training and testing of the algorithms, we employ a methodical data filtering process. The process is specified in Figure 7.

All datasets go through the following three steps (see Figure 7): check for duplicates, aspect ratio anomaly check, and RGB anomaly

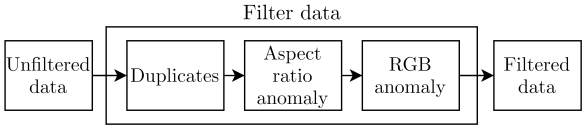
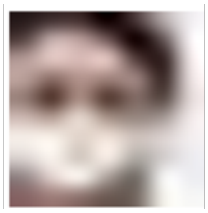


Figure 7: Data Filter Process

check. The duplicate check is accomplished by a method proposed by Li et al. (2016). Li et al. (2016) propose calculating the perceptual hash of each image and comparing these hash values against each other to find duplicates. The perceptual hash function is reasonably lightweight, making it feasible to calculate the hash value of the approximately 40,000 images. In total, approximately 400 duplicates are found and removed within and across the datasets.

The next step in the filtering process is detecting aspect ratio anomalies. Since all images at a later point have to be resized into an aspect ratio of 1, some images and their features become too stretched out. This tends not to be a problem for larger images. Therefore, images with a square root size under 300 pixels and an aspect ratio outside the span of 0.25-4.0 have been removed. Figure 8 shows a sample of two images removed in this filter.

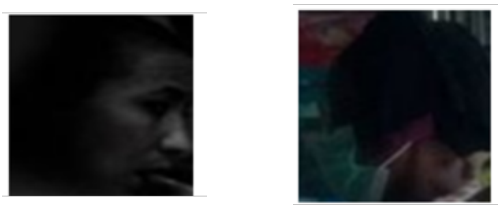


(a) Removed due to Low Size

(b) Removed due to High Aspect Ratio

Figure 8: Low Resolution Images and Bad Aspect Ratios

The final step is detecting anomalies in RGB values by calculating each average of red, green, and blue values. All pictures having less than five or greater than 250 in any of the average values are filtered out. This primarily detects pictures that are almost completely black or white. Figure 9 displays two images removed due to low RGB values. There were no images that were too bright.



(a) Removed due to Low RGB Values (b) Removed due to Low RGB Values

Figure 9: RGB Anomaly Sample

A complete overview of the distributions of size and aspect ratio of every dataset and class can be found in Appendix A.

4.2.2 Data Normalisation

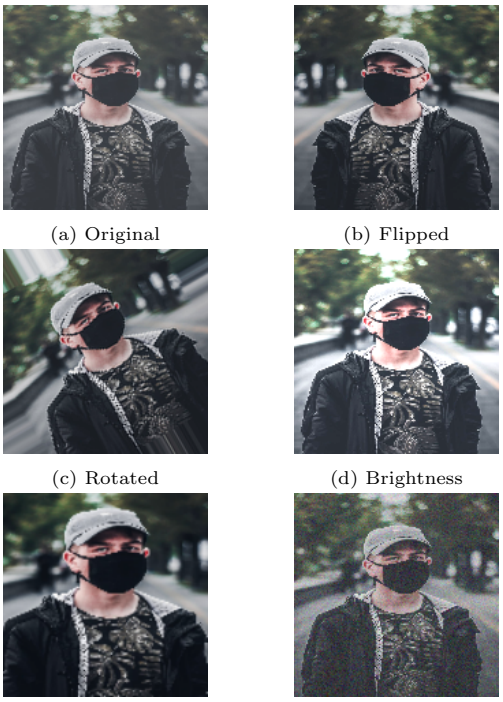
Normalising the data is a crucial step to create consistent datasets. As explained in section Dataset Description, the height and width of the images vary between and, in some cases, within the datasets. To normalise the size of the images, all images have been transformed into the resolution 150x150. This size is sufficiently small to make it feasible to train algorithms and large enough to contain details that an algorithm can use to differentiate the two classes. Furthermore, all pixel values have been rescaled to have values between 0 and 1.

Finally, at the end of the data normalisation process, we combine the data sources into one and split the data afterwards. Dataset 1 is isolated and split alone, while dataset 2.1, 2.2, 3.1, and 3.2 are combined before the split. We decided to split the data such that the training set contains 60% of the images, and both the validation and test set each contain 20%. Dataset 1 is split into train, validation, and test using the same ratio.

4.2.3 Data Augmentation

To perform a more robust training and validation of the models, we choose to augment the data. By adding modified images, we increase the data size and the models' ability to generalise. The following methods have been used to increase the number of data points: flip, rotate, brightness, crop, and noise. Dataset 2.1 was not augmented because it already contained augmented images.

First, the images have been flipped from left to right and rotated by 30 degrees, making it possible to capture different angles of the faces. Besides that, the brightness of the images has been randomly adjusted to account for different lighting conditions. To obtain images that are more focused on the faces, the original data has been cropped such that 80 % of the image is preserved. Furthermore, Gaussian noise has been added to all images to increase the variability of the data. The transformations of each image can be seen in Figure 10.



(a) Original (b) Flipped (c) Rotated (d) Brightness (e) Cropped (f) Gaussian Noise

Figure 10: Augmented Images - Dataset 2.2

The size of the different training datasets, both original and augmented, is presented in Figure 11 and Figure 12.

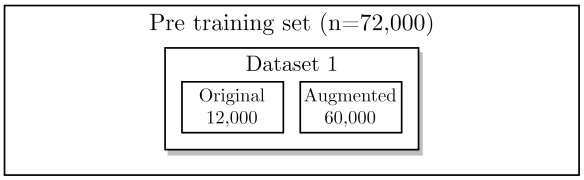


Figure 11: Pre-training Set for the CNN that uses Transfer Learning

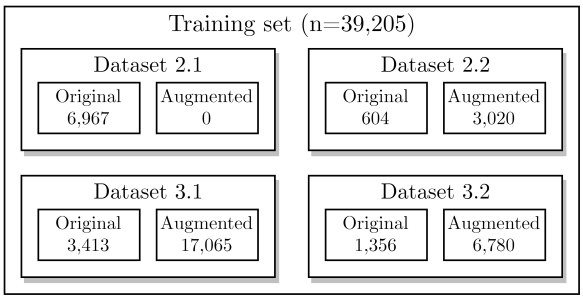


Figure 12: Distribution of Training Data Except Dataset 1

4.3 Modelling Framework

In this project, we will deploy four different models. Two custom-designed CNNs that are based on the AlexNet architecture. One of which will be pre-trained and used in a transfer learning setting where more layers are added. The second will be the same fully layered CNN, but it will not be pre-trained. We will benchmark these models against each other and two other models: a pre-trained InceptionV3 model and a Support Vector Machine.

4.3.1 Convolutional Neural Networks

We will test out a convolutional neural network with transfer learning to classify whether a person wears a mask or not. See Figure 2. We hypothesise that by training a CNN on images where the masks are artificially made (see Figure 11), we can obtain trained weights that can be transferred to another CNN model, which makes it easier to detect masks. By training on the pre-training set, we believe that the model's weights will converge faster as the images in the pre-training set are more homogeneous than the real training set. After the pre-training, we will deploy the model on the training dataset and add some more layers. This is done in the hope of improving generalisability.

The benefit of using pre-trained weights is that the model can use an established knowledge about people with and without masks when the amount of training data is limited. The base model employs several convolution and pooling layers to extract and summarise the features of the images. Moreover, two regularisation methods, batch normalisation and dropout have been used. Batch normalisation is added to the model architecture to reduce the vanishing gradient problem and stabilise the input distribution to a layer in the network. Dropout layers are used to prevent the model from overfitting.

After having applied both convolution and pooling layers, the data is flattened in order to add a dense layer to extract a logit value, representing whether or not the person in the image wears a mask or not. In the hidden layers, we decided to use Rectified Linear Unit as it has favourable attributes with regards to vanishing gradients in deep neural network architectures (Charu, 2018, Chapter 3). Furthermore, we have used an Adam optimiser with a learning rate of 0.0001 and a binary cross-entropy loss function. The network was trained for 15 epochs with 64 images per batch and preserving all three colour channels.

After the base model is trained, the weights are transferred to another model that performs the same classification task using real masks instead of artificially made masks. In this case, we decided to add some additional layers to the pre-trained base model to detect more detailed features of the images. The training of this new model is split into two phases. First, the model is trained using only one epoch to initialise the weights of the new layers. The weights from the base model are frozen such that it is only the new weights that are trained. In the second phase, all the layers in the network are trained together to optimise the model's performance. The model architecture is visualised in Appendix B.

To explore the robustness of transfer learning, a regular CNN is used for comparison. This model has the same architecture as the model used for transfer learning to make a more reasonable comparison. The only difference is that the model is trained and validated on all images except dataset 1. The model architecture is visualised in Appendix B.

4.3.2 ImageNet InceptionV3

To reason about the upper limits of the dataset, we train another model based on the inception architecture. The inception architecture is based on the work of Szegedy et al.

(2015) and is centred around the idea that crucial information in the images is captured at different levels of detail. In traditional CNN, if we use a large filter, we can capture information in a bigger area. If we use a smaller filter, we capture information in a smaller area. The appropriate filter size might vary between images, and therefore, we need to adjust for this problem. More technical information about the inception architecture can be found in Appendix C.

In our paper, we will use an ImageNet pre-trained version of InceptionV3 to benchmark our other methods. Extracted features from image data based on ImageNet should be highly reusable across different problem domains. ImageNet is an image dataset containing more than 14 million images annotated with more than 20,000 different labels. Training on this generic data source, the InceptionV3, will capture the primitive features in the earlier layers. We will transfer this pre-trained InceptionV3 model into our problem domain, switching out the last multi-class classification layer into a binary classification layer.

When training InceptionV3, we train it in two stages: First, we train only the newly added last layer for a single epoch to stabilise its weights. This ensures that the weights in the last layer do not fluctuate too much between training. Second, we train the model again, but with all the weights and a lower learning rate.

We choose this model because of the computational efficiencies that it provides, making it easier to train and because it has proved to be an excellent model for image classification problems Szegedy et al. (2016). We hypothesise that the ImageNet-InceptionV3 model will do quite well on this mask classification problem compared to the other models in this paper.

4.3.3 Support Vector Machine

To compare the effect of using a non-neural approach with the neural approaches, we use a support vector machine (SVM) classifier. The idea of the model is to define a hyperplane that makes it possible to distinguish the two classes of images from each other. The optimal solution is a plane that maximises the margin between the data points from both classes, which can be used as a decision boundary.

Before training the SVM model, a randomised principal component analysis is applied to reduce the number of features. By using 100 components, we preserve around 87% of the variance. To get the most suitable set of parameters in the SVM model, we have performed a grid search, incorporating a 5-fold cross-validation. Based on the grid search result, we choose the radial basis function kernel together with the suggested values for the other tested parameters.

4.4 Evaluation Metrics

We choose to evaluate our models based on the F1-score on the test set, which is the harmonic mean between precision and recall. The F_1 score consists of the precision score and recall score, which is calculated by:

$$Precision = \frac{True\,positives}{True\,positives + False\,positives}$$

$$Recall = \frac{True\,positives}{True\,positives + False\,negatives}$$

The F_1 score is calculated by:

$$F_1 = \frac{2}{Recall^{-1} + Precision^{-1}}$$

Precision and recall are a trade-off of each other. One can achieve a higher recall by having more false positives, but this will reduce the precision, and the reverse is true. The F_1 score considers this trade-off. Maximising the F_1 means maximising both the precision and recall.

To calculate the aggregated performance measure of both of the labels, we will mainly focus on looking at the macro-average of the F_1. The problem with the micro-average is that for unbalanced datasets, the label with the larger number of data points will be given a larger weight. We want to give equal weight to each of the two labels. Since our data is slightly unbalanced, the macro-average will work more favourably as we have more images with masked people.

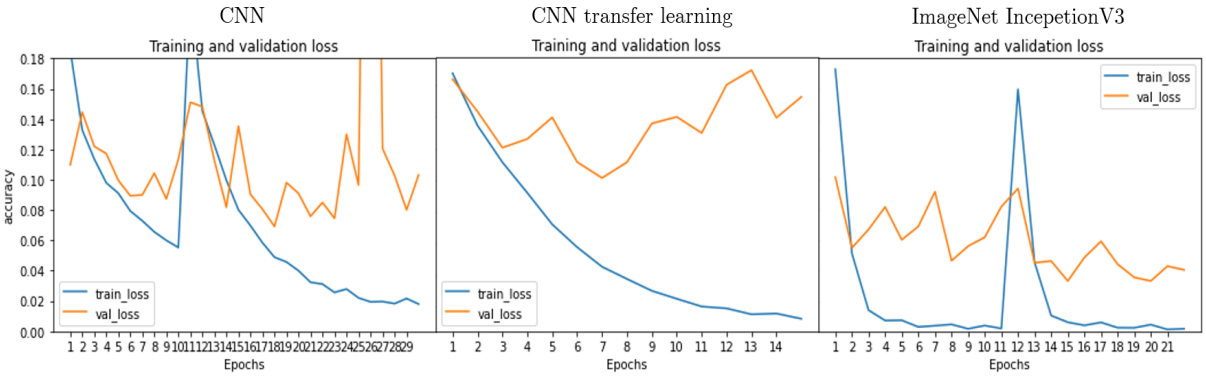


Figure 13: Training Loss Results

5 Results

The test results are clear. The best performing model is the InceptionV3 with an F_1 score of 0.993 on the test set. The CNN model without transfer learning follows it with an F_1 score of 0.981, then CNN with transfer learning with an F_1 score of 0.969, and finally SVM with an F_1 score of 0.927. The details of the precision, recall, and F_1 scores can be found in Table 3.

The test set consists of 4,102 faces, where 2,448 accounts for masked images and 1,654 accounts for non-masked images. This might be why we see a lower value of precision and recall for no-masked labels across the models. The confusion matrices can be found in Appendix E. They show the predictions of each label against the actual label value. We use it to see how many false negatives and false positives each model has on the test set. We note that there is a higher value of false negatives. This means that the models have a larger tendency to classify non-masked images as masked images. Again, this is probably caused by the unbalanced dataset.

For the CNN and InceptionV3 models, we present the training loss in Figure 13. For the CNN with no transfer learning, we see that after around 20 epochs, the model begins to overfit as the training loss and validation loss become increasingly uncorrelated. For the CNN transfer learning model, we see that the training and the validation loss changes more smoothly, although the model tends to overfit earlier and the spread between the training and validation loss is larger than the other

CNN no transfer learning ($F_{1_{Macro}} = 0.98$)			
Label	**Precision**	**Recall**	**F_1-score**
Mask	0.99	0.98	0.98
No mask	0.97	0.98	0.98
CNN with transfer learning ($F_{1_{Macro}} = 0.97$)			
Mask	0.98	0.97	0.97
No mask	0.96	0.96	0.96
ImageNet InceptionV3 ($F_{1_{Macro}} = 0.99$)			
Mask	0.99	1.00	0.99
No mask	0.99	0.99	0.99
Support Vector Machine ($F_{1_{Macro}} = 0.93$)			
Mask	0.94	0.94	0.94
No mask	0.91	0.92	0.91

Table 3: Precision, recall, and F_1 scores

models. For both the CNN with and without transfer learning, we see that the model probably has reached its optimum training and that training for more epochs will probably not increase their performances. Looking at the pre-trained InceptionV3, we see that the training loss reduces at a high rate in the first 2-3 epochs. Already after three epochs, the InceptionV3 model reaches a lower validation and training loss than both of the CNN models after their last epochs. Interestingly, the InceptionV3 model might not have reached its optimum at the last epoch yet, as it looks like the validation loss is still trending downwards. The training and validation accuracies can be found in Appendix D.

5.1 Complexity & Running Time Analysis

When assessing the quality of a model, it is also essential to consider the training time of the models because sometimes a less accurate model might be preferable because of a lower running time. All the models have been executed on a machine with 371 GB RAM and 63 vCPUs. The running time of the different models in this paper is visualised in Figure 14. The figure shows that the CNN transfer learning model has the longest running time compared to the other models. The reason is that we first have to train the base model on a large dataset, and afterwards, the pre-trained weights are transferred to another model that also requires training. By looking at the F_1 score and the other models' running times, it is possible to argue that this model is not the most effective one. Instead, it is more suitable to use the InceptionV3 model as it is 1.5 and 2.6 times faster than the CNN and CNN with transfer learning, respectively. Considering the F_1 score, the model also outperforms the other models. It is important to note that the running times of the neural models are highly dependent on the number of epochs required for the models to converge.

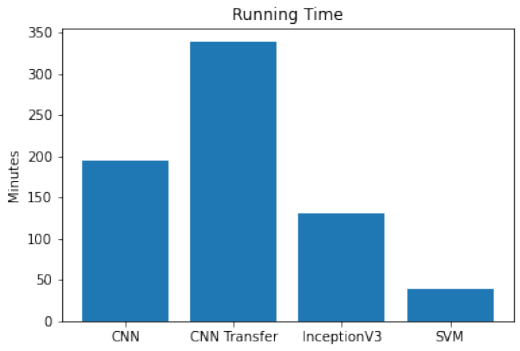


Figure 14: Running Times

When looking at the SVM, it completely outperforms the other models in running time even though it performs a grid search cross validation. The model is 3.4, 5, and 8.7 times faster than the InceptionV3, CNN, and CNN with transfer learning, respectively. As a result, it is essential to consider the trade-off between performance and running time because SVM achieved the lowest F_1 score, but it is by far the fastest model.

6 Discussion

6.1 Comparison of the Models

By looking at Table 3, it is clear that the neural models outperform the SVM. As this paper mainly focuses on using neural models for image classification, the optimisation performed concerning the SVM is limited. If we have invested more time and resources in optimising the SVM, there would have been a higher chance of reaching the same F_1 score as the neural models. When focusing more on the neural approaches, it is evident that the CNN that uses transfer learning lacks a bit of performance compared to both the InceptionV3 and the regular CNN model. Figure 13 clearly shows that the gap between training and validation loss is larger than for the two other models, which indicates that the model tends to overfit more. The reason for this can be found in the training of the base model. The dataset used for training the base model contains white artificially made masks, and the variety in the images is limited. This makes the weights of the pre-trained model biased towards these features, and we theorise that these biases tend not to be nullified even after training for a vast number of epochs. Instead, the weights of the newly added layers will be more prone to "memorise" the generated features of the previous layers, making them overfit rather than generalise.

By increasing the variety and size of the training data in the base model and more robust training similarly to the InceptionV3 uses ImageNet, it might be possible to increase the model's performance. For future work, since we are primarily interested in human heads and faces, it might be interesting to pre-train a model that specialises in recognising faces or heads. We hypothesise that this model would be able to pick up the human facial features and, therefore, perform a better holistic evaluation. The data needs to be very diverse to avoid the overfitting tendency, as seen in this project's pre-trained CNN.

6.2 Error Analysis

Out of 4,110 images, the CNN model had 76 wrong predictions, the CNN transfer model had 123 wrong predictions, the InceptionV3 had 27 wrong predictions, and lastly, the SVM had 288 wrong predictions. In this section, we will look into some of these wrong predictions and explain why each of the models failed.

We see that many of the wrong predictions are because of lousy image data. If we take a look at some of the wrong predictions of InceptionV3, the best predicting model, we see the following images in Figure 15.

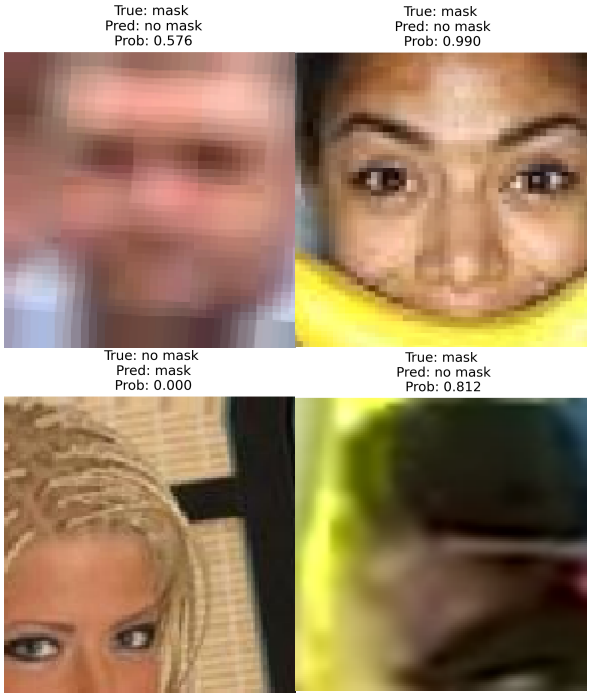


Figure 15: InceptionV3 Miss-classified Images. ("Prob" is the probabilities from logits values of having no mask. If prob ≥ 0.5, then the model classifies the image as having no mask.)

Even for regular people, it would be hard for them to distinguish whether these images are labelled as masked or not. The upper left image is too blurry. For the woman in the upper right, it looks like she has a banana as a "mask". In the lower-left, the woman's lower face is cropped out. In the lower right, it looks like someone is wearing a hazmat suit, not showing any part of his/her face. Other errors include wrongly labelled images. See Figure 16.

Figure 16: Mask or no mask?

Apart from these errors, the InceptionV3 only misses predictions on images where the face cannot be seen. Typically these images are of the back of people's heads.

Looking at the errors of the CNN transfer model, we notice a somewhat worrisome trend regarding colours. It errors more often on images of people with darker skin with the false label of having a mask, even when the picture clearly shows them having no mask. The CNN transfer model is first trained using a dataset with only white masks. Therefore, the weights might be biased toward looking for light parts of the image in order to establish whether someone wears a mask or not.

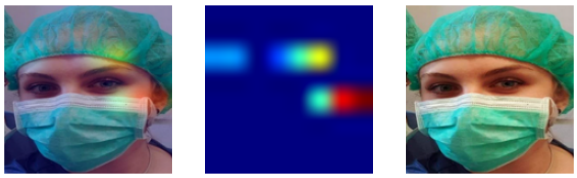
6.3 CNN Layer Analysis

In this section, we will analyse the CNN model trained without transfer learning. To get a better understanding of this model, we carry out a Grad Cam analysis (Selvaraju et al., 2017). Other visualisation techniques that we could have used include guided backpropagation (Springenberg et al., 2014) and deconv (Zeiler and Fergus, 2014). We favoured Grad Cam because of its simple interpretability and easy to read visualisation. Usually, Grad Cam is used in multi-class classifications, but since we have a binary classification problem, we need only to analyse one of the classes to determine how the model classifies.[2]

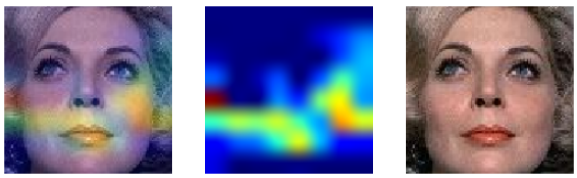
The last layer in our CNN consists of a single neuron that outputs a logit: if the logit value is above 0, the model labels the image

[2]Since we have two classes, we have one degree of freedom. Determining whether a label is positive or negative determines the other class as well.

as having no mask. If the logit value is below 0, the model predicts the image as having a mask. This means that our model is not built to recognise when a mask is present. Instead, it is built to recognise whether there is no mask present. The effect of Grad Cam is that it "lights up" only when it detects features that it connects to having no mask. We expect Grad Cam to find close to none or only weak features for masked images, and for non-masked images, we expect Grad Cam to light up the features, which it finds relevant.



(a) Prediction Label: With Mask
Logit Value: -146



(b) Prediction Label: Without Mask
Logit Value: 143

Figure 17: Left: Grad Cam, Mid: Heatmap, Right: Original Image. Showing the last convolutional layer for the images with the lowest and highest logit value in the training set. Model: CNN without Transfer Learning.

In Figure 17, we see two images chosen based on the predicted label. The upper image has the lowest logit value of all images in the test set. The lower image is the image with the highest predicted logit value. This means that the model is most particular about whether these images are masked or not. We see for the masked image that the model finds some features belonging to the non-masked class but not enough features to classify it as non-masked. Since the model wrongly finds features related to the non-masked label, the model can still be improved, as a perfect model should not pick up on these false positive features. Looking at the lower set of images, we see a non-masked labelled image. Here, the model has focused on the area near the mouth and the cheeks. Incidentally, across many of the correctly labelled images of no-masked people, we see that the CNN model tends to look at these places. Some more Grad Cam images can be found in Appendix F.

To conclude, it seems that the model can pick up some reasonable patterns in order to determine whether the person in the image wears a mask or not. To determine this, the model looks for cheeks, ears or chins/mouth. If it cannot find those characteristics, then the model determines that the person in the image wears a mask. Concurrently, we find that the model is not perfect and that it sometimes picks up false patterns. This might be solved by having a bigger training set or having a more complex model. Furthermore, we did not do this analysis on the pre-trained InceptionV3 model. Doing the same analysis on the InceptionV3 model might show the limits of the training set and give some clues of the quality of the data.

7 Conclusion & Future Work

In this project, we trained four models to classify whether a person in an image wears a mask or not. We built two CNNs, one where we tried to utilise transfer learning and another one without. We see that the one with transfer learning is performing worse because of the poor quality of the pre-training dataset, which is best described as homogenous. Furthermore, we trained an SVM model, acting as a lower benchmark, and a pre-trained InceptionV3 model, which we used as an upper benchmark. Throughout all of these models, we achieved an average F_1 score of 0.96. The best model, the pre-trained InceptionV3, achieved an F_1 score of 0.99, making it very powerful concerning this classification task.

Finally, we have not discussed mask classification's ethical concerns and societal impact, leaving this as future work. A model being able to classify images of masked or non-masked persons can be used by governments to obtain statistics to understand masked prevention in connection to diseases better. However, it might also be abused in politics by governments that want to ban religious head-

wear. Furthermore, before deploying such a model, more meta-information about the training data is needed to ensure that the model is not only biased in terms of the labels in question but also whether it is biased in terms of race, colours, sex, and so on.

References

C Aggarwal Charu. 2018. *Neural Networks and Deep Learning*.

G Jignesh Chowdary, Narinder Singh Punn, Sanjay Kumar Sonbhadra, and Sonali Agarwal. 2020. Face mask detection using transfer learning of inceptionv3. In *International Conference on Big Data Analytics*, pages 81–90. Springer.

Benjamin J Cowling, Kwok-Hung Chan, Vicky J Fang, Calvin KY Cheng, Rita OP Fung, Winnie Wai, Joey Sin, Wing Hong Seto, Raymond Yung, Daniel WS Chu, et al. 2009. Facemasks and hand hygiene to prevent influenza transmission in households: a cluster randomized trial. *Annals of internal medicine*, 151(7):437–446.

Xuan Li, Jin Li, and Faliang Huang. 2016. A secure cloud storage system supporting privacy-preserving fuzzy deduplication. *Soft Computing*, 20(4):1437–1448.

Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N Taha, and Nour Eldeen M Khalifa. 2021. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic. *Measurement*, 167:108288.

Bosheng Qin and Dongxiao Li. 2020. Identifying facemask-wearing condition using image super-resolution with classification network to prevent covid-19. *Sensors*, 20(18):5236.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

K Suresh, MB Palangappa, and S Bhuvan. 2021. Face mask detection by using optimistic convolutional neural network. In *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pages 1084–1089. IEEE.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

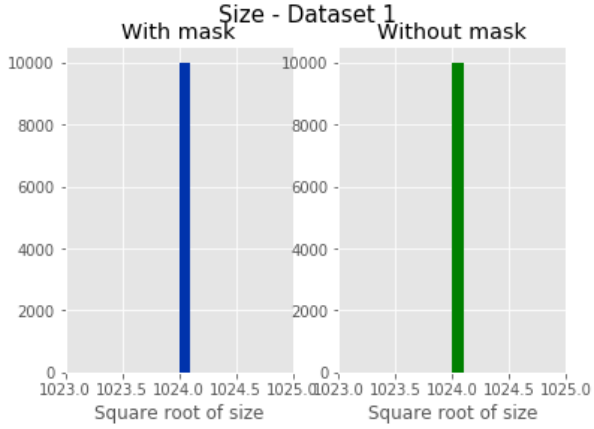
Samantha M. Tracht, Sara Y. Del Valle, and James M. Hyman. 2010. Mathematical modeling of the effectiveness of facemasks in reducing the spread of novel influenza a (h1n1). *PLOS ONE*, 5(2):1–12.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.

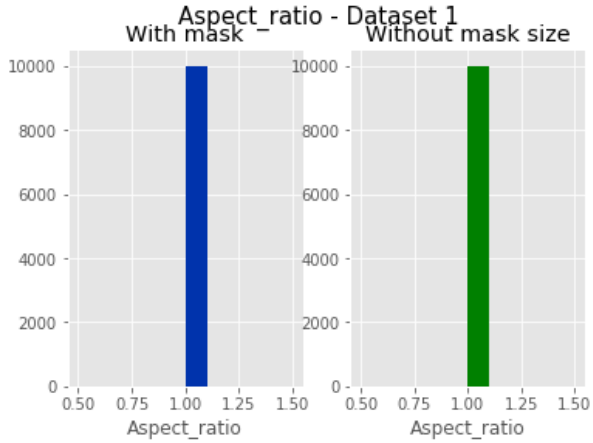
Appendix

A: Data Exploration

A.1: Dataset 1

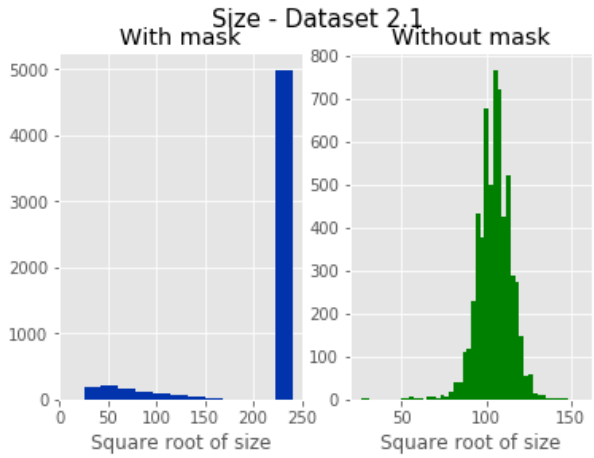


(a)

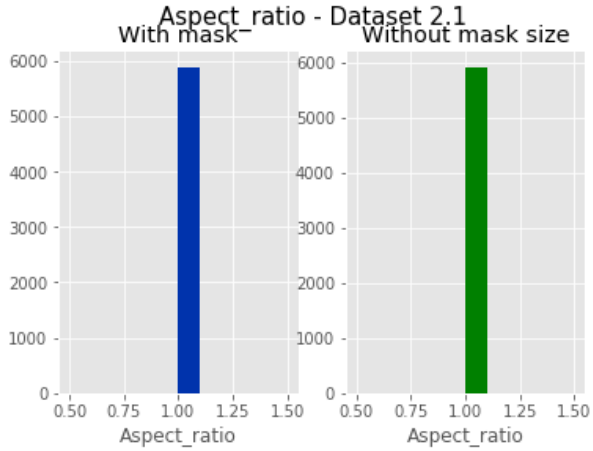


(b)

A.2: Dataset 2.1

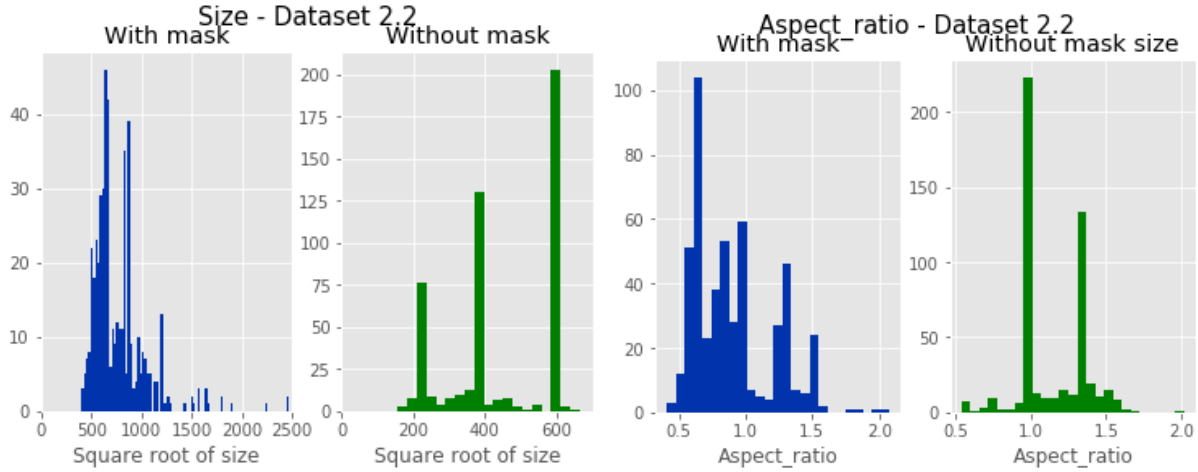


(a)



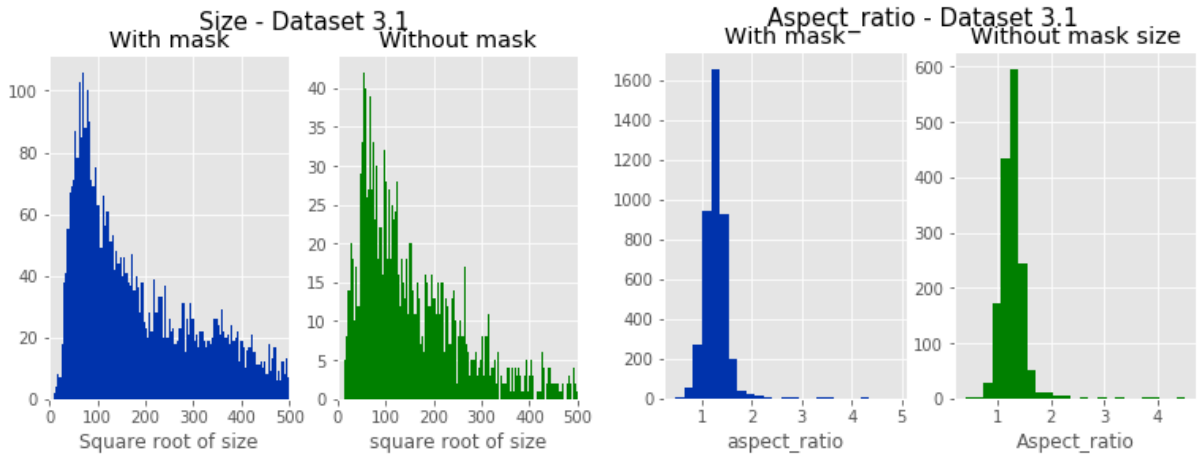
(b)

A.3: Dataset 2.2



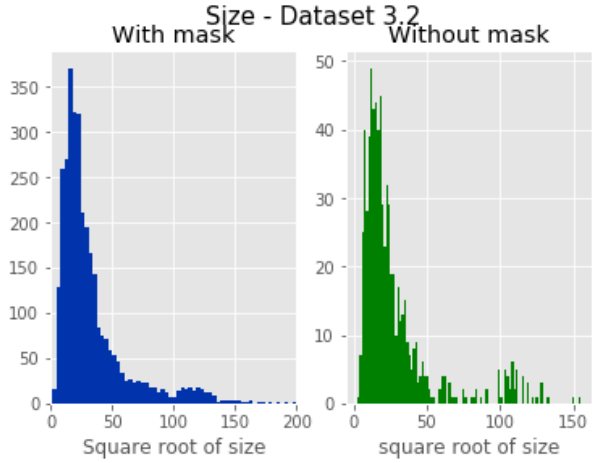
(a) (b)

A.4: Dataset 3.1

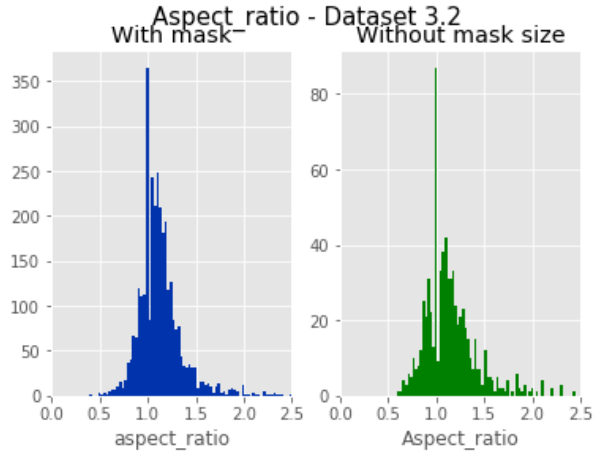


(a) (b)

A.5: Dataset 3.2

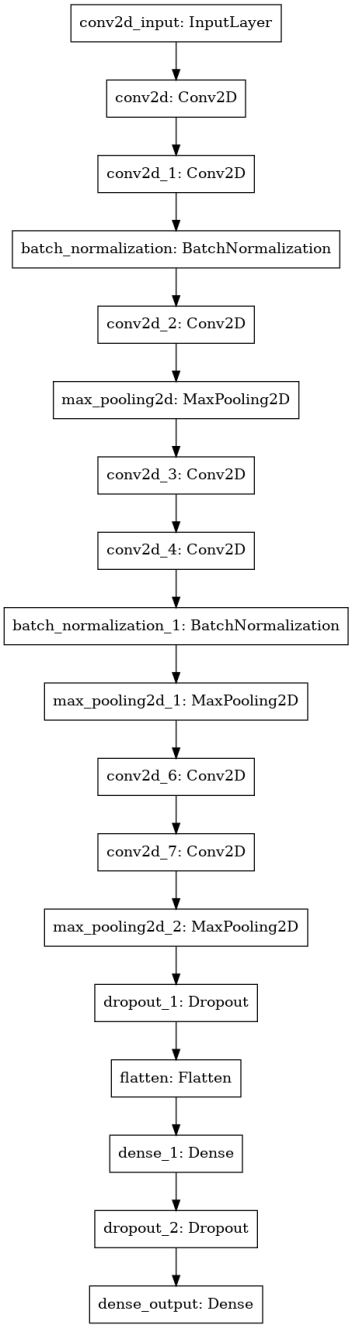


(a)



(b)

B: Convolutional Neural Network - Model Architecture



C: Imagenet InceptionV3

The inception module solves this by using a network-within-network architecture, where the inception module consists of several convolutional layers in parallel, each with different filter sizes. See Figure 23.

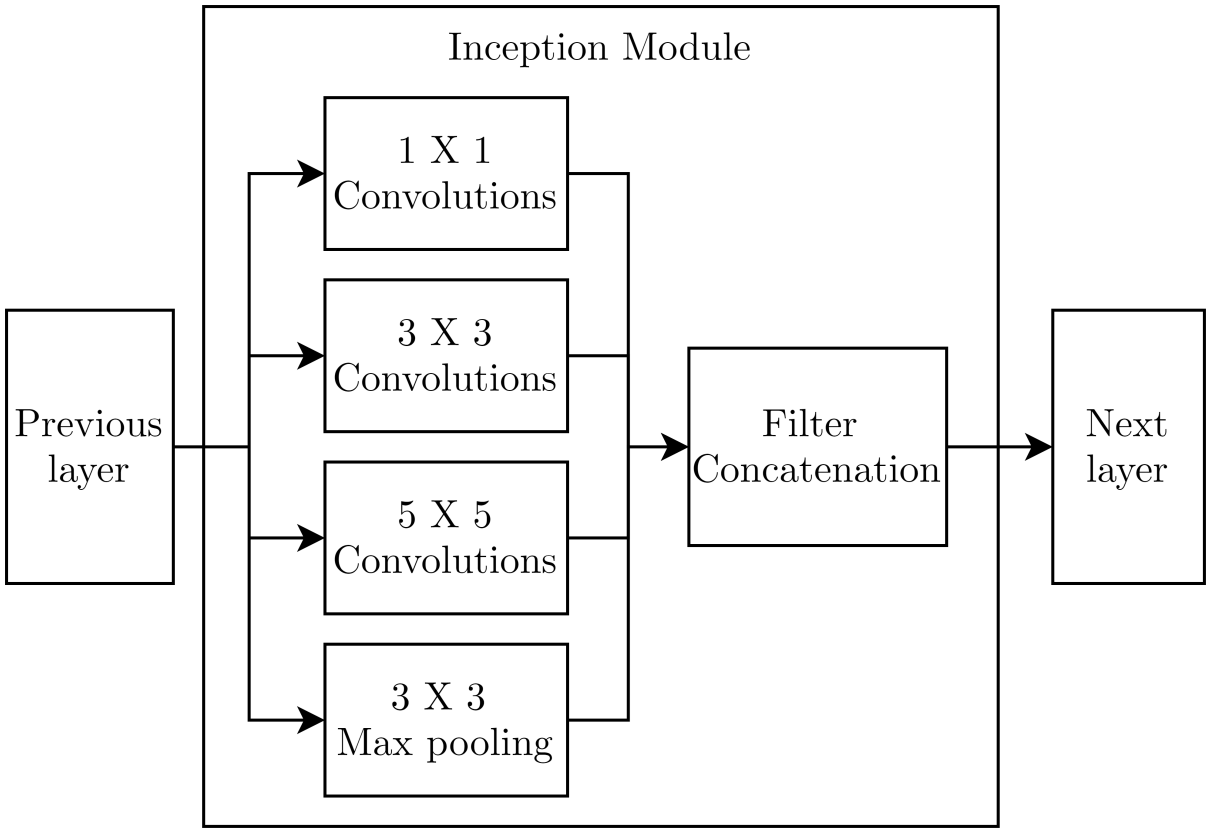
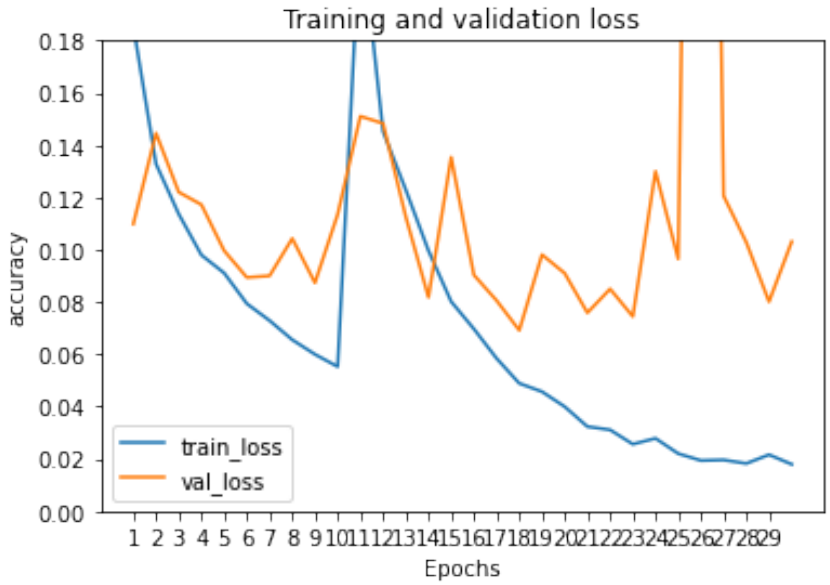


Figure 23: Inception architecture

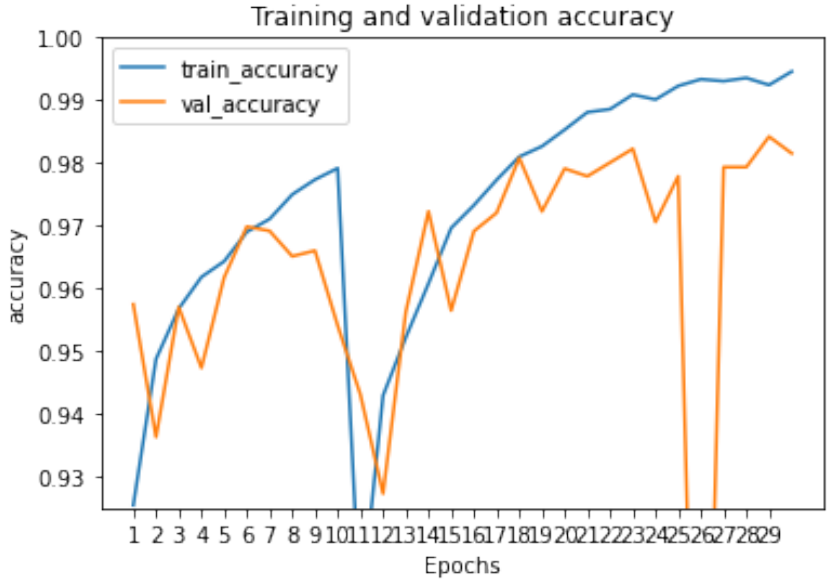
InceptionV3 is a particular incarnation of the inception architecture that optimises the computation by factorising the convolutions in the inception module. Szegedy et al. (2016) argue that by transforming a $n \times n$ convolution into a $1 \times n$ convolution followed by a $n \times 1$ convolution, the computational efficiency cost saving grow as n grows.

D: Training Results

CNN Training Results

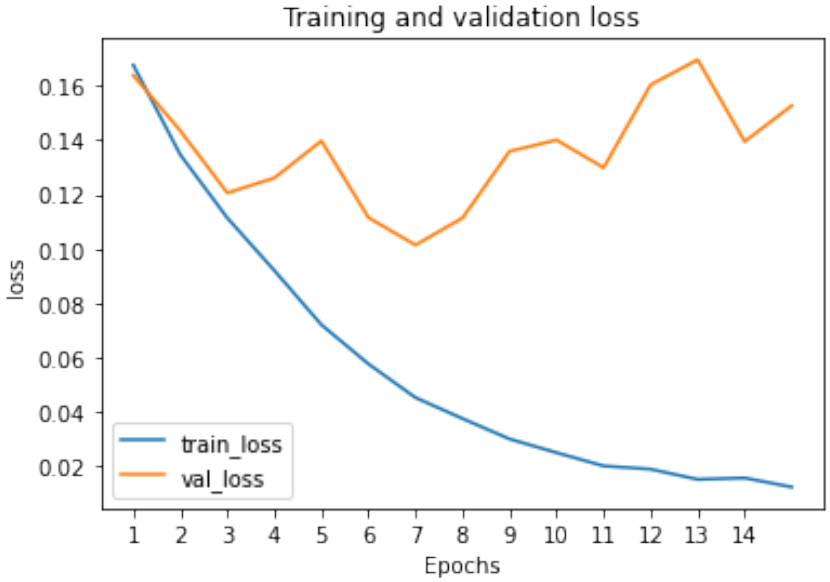


(a) Loss on training and validation set

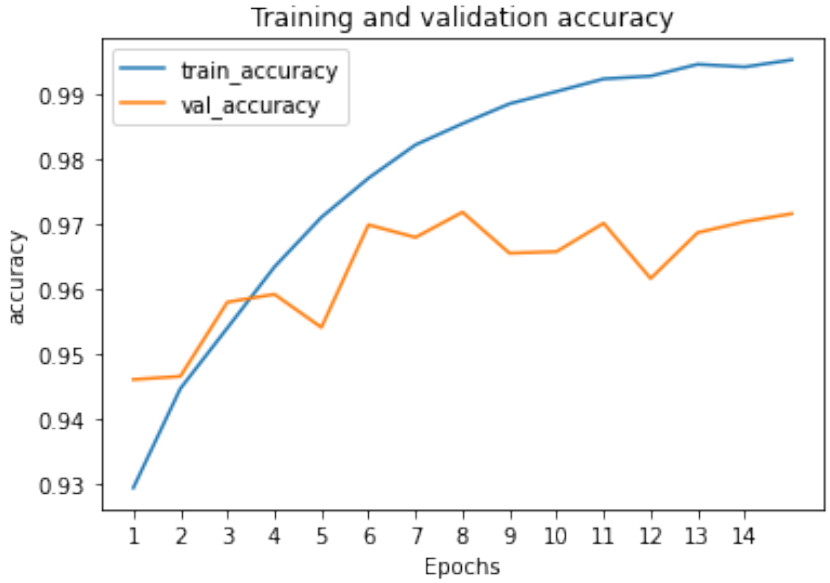


(b) Accuracy on training and validation set

CNN Transfer Model Training Results

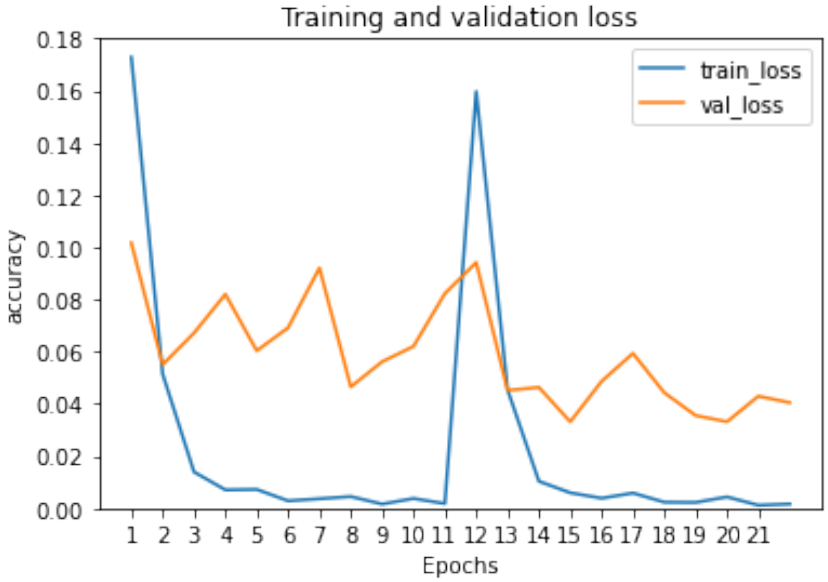


(a) Loss on training and validation set

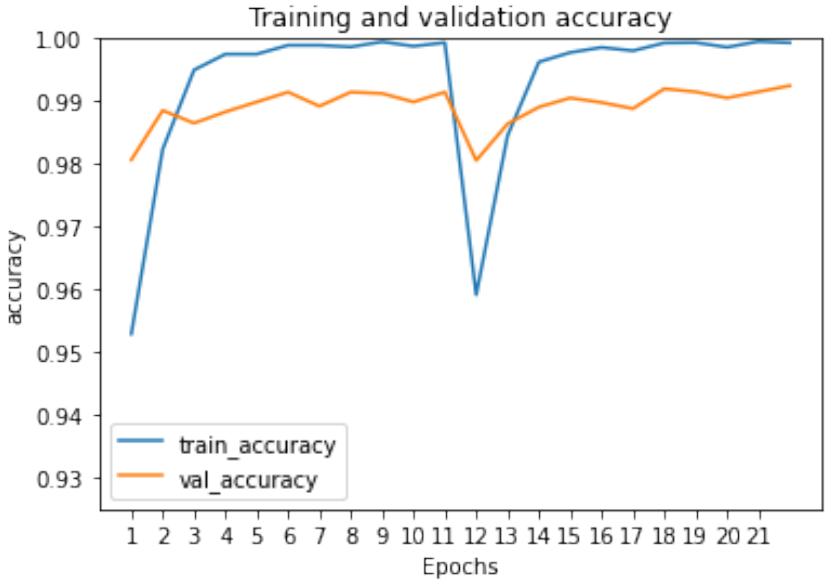


(b) Accuracy on training and validation set

ImageNet InceptionV3 Training Results



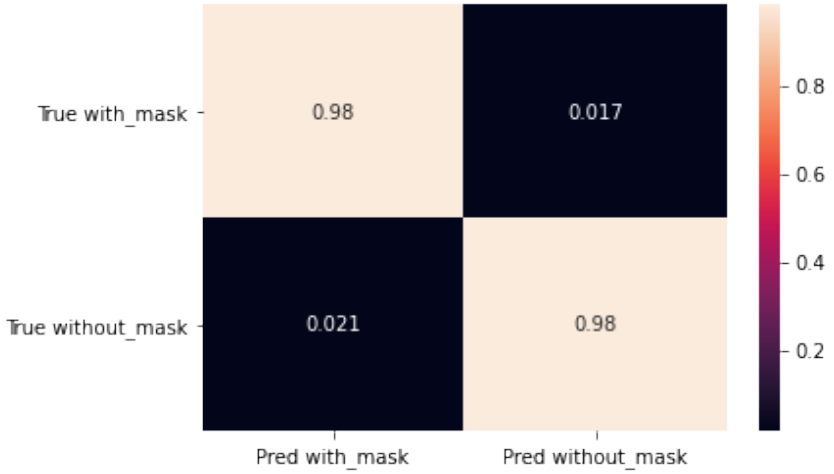
(a) Loss on training and validation set



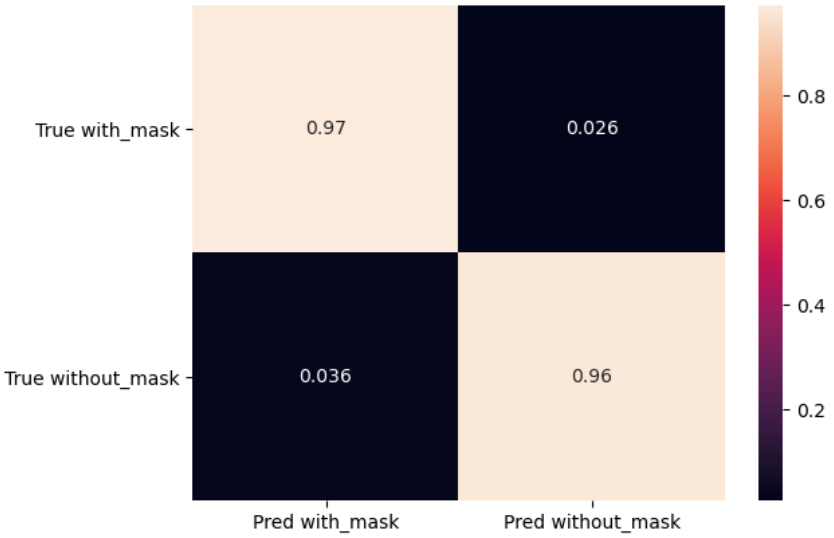
(b) Accuracy on training and validation set

E: Confusion Matrices

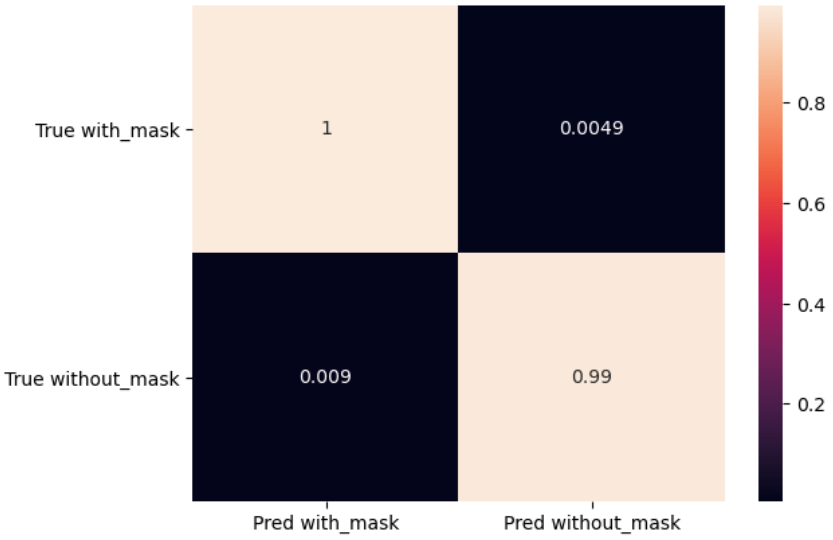
CNN Confusion Matrix



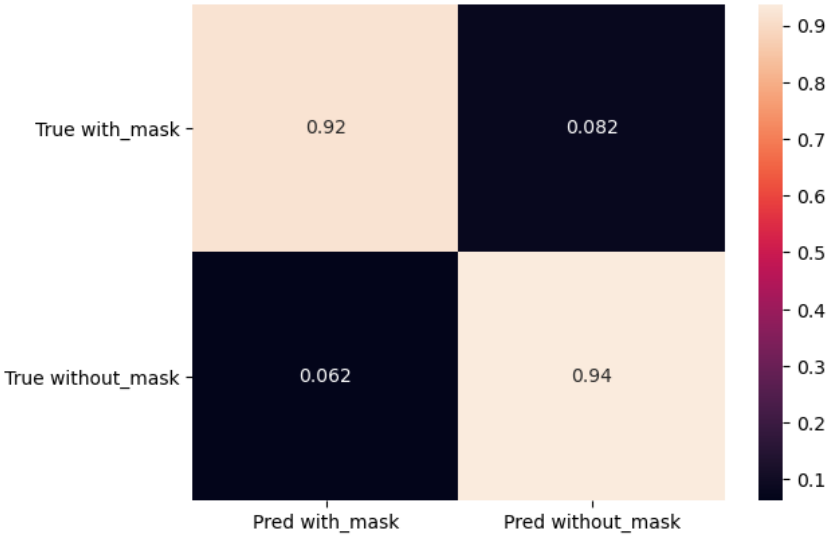
CNN Transfer Model Confusion Matrix



ImageNet InceptionV3 Confusion Matrix



Support Vector Machine Confusion Matrix



F: Grad Cam - Without Mask