Computer Vision report

Jessica Guan

Data

The provided data to train and test the models consisted of 2394 train images and labels and 458 test images and labels. There were 3 unique classes: 0 (no mask), 1 (mask), 2 (mask improper). As shown in Table 1, the data was heavily imbalanced, with class 0 as the majority class. As shown in Table 2, the range of image sizes was large, with 44.86 x 40.3 as the average size of the training images. The video used to test the best trained model was recorded by me with myself as the tester. The video is about a minute long of myself showcasing no mask, mask, and mask improperly worn.

Table 1: Class distribution

	Train	Test
Class 0	376 (15.71%)	51 (11.14%)
Class 1	1940 (81.03%	388 (84.72%)
Class 2	78 (3.26%)	19 (4.15%)

Table 2: Training image sizes

20x20 and below	350
20x20 to 50x50	1162
50x50 to 100x100	261
100x100 to 200x200	144
200x200 and above	477

Implemented methods

This project includes 5 computer vision models: Resnet-50, Mobilenet V2, custom CNN, histogram of features (HOG) + support vector machine (SVM), and Scale-Invariant Feature Transform (SIFT) + support vector machine (SVM). For all the models, the train images and labels were split into 80% training set and 20% validation set.

HOG + SVM

I chose HOG + SVM to test because it has capabilities to capture objects and textures which could be useful in face mask detection and facial features. Data augmentation was applied to provide more variability for the model to learn. Because the training set was imbalanced, I experimented with oversampling. However, oversampling yielded worse results in the validation set, so I proceeded with the original class distribution in the training set. For training, validation, and testing, the images were converted to grayscale, resized to 64 x 64, and the pixels were normalised. The resized image size was found through manual testing. Then, the HOG features were extracted. In the HOG function the orientation, pixels per cell, and cell per block best values were also found through manual testing. To find the best parameters, GridSearchCV from sklearn was used to iterate through the C parameter, kernel type, and kernel coefficient, in which cross-validation was performed internally within grid search. With the best parameters found through grid search, the HOG + SVM model was trained on the training set and validated on the validation set. Lastly, the trained model was tested on the test image and labels.

SIFT + SVM

I chose SIFT + SVM because of its ability to detect keypoints and descriptors, a useful method for facial features. The training images were converted to grayscale, then SIFT was applied to the images to extract and append the features to an array. Because RBF was the best kernel type for HOG + SVM and it is useful for complex and nonlinear data patterns, RBF was also used as the kernel type for SIFT + SVM. Iterating through values for the C parameter and kernel coefficient, the best parameters were found through sklearn's GridSearchCV, in which cross-validation was performed internally within the function. With the best parameters, the SIFT + SVM model was trained on the training set. After using the same preprocessing steps as the train images on the validation and test set, the trained model was validated and tested on the validation and test set respectively.

Table 3: Class distribution in training set after oversampling

Class	Count
Class 0	1880 (33.03%)
Class 1	1940 (34.08%)
Class 2	1872 (32.89%)

Resnet-50

Resnet-50 has a deep architecture and was trained on a large dataset so it can learn complex image features. The train, validation, and test images were resized to 224 x 224, as this is the image size that Resnet-50 expects. The images were also normalised and transformed into tensors. Because of the data imbalance, oversampling and data augmentation was applied only to the training set. After this, the train, validation, and test sets were put in data loaders called train_loader, val_loader, and test_loader respectively. A baseline Resnet-50 model was trained and tested to observe the results. Then, a combination of grid search and manual testing was used to find the best parameters and architecture. With the best parameters and

architecture, the model was trained, validated, and tested on the training set, validation set, and test set respectively.

Mobilenet V2

Like Resnet, Mobilenet is able to learn complex image features, but it is more lightweight and computationally efficient. The same preprocessing, data augmentation, and oversampling steps for the train, validation, and test images as Resnet-50 were applied for Mobilenet V2. Also, like for the Resnet-50 model, a combination of grid search and manual testing was performed to find the best parameters and architecture. With the best parameters and architecture, the model was trained, validated, and tested on the training set, validation set, and test set respectively.

Custom CNN

A custom CNN allows for flexibility and experimentation to tailor the model specifically for face mask detection. The train, validation, and test images were resized to 224 x 224. Using Keras, a baseline model was built using code reference from Kaggle [1]. The model was adjusted for multi-classification. The architecture was tweaked and different values for pooling, dropout, number of neurons were used. An additional linear layer was added. These changes were a result of experimentation for the best results. I applied the model to the original imbalanced train data and the oversampled data. The model's validation results showed better results on the oversampled data, so the CNN model was trained on this. Then, it was validated on the validation set, and tested on the test set.

Results

The results of the classification reports are shown in Table 4 (macro averages) and Table 5 (weighted averages). HOG + SVM is a classic computer vision model. Although it shows high accuracy and precision scores, the recall and F1 score is lower. This indicates that the model struggled in classifying all the true positives. The results show that the model had difficulty capturing the relevant HOG features, possibly due to the data complexity.

Table 4: Macro averages classification report

	Accuracy	Precision	Recall	F1
HOG + SVM	0.91	.91	.60	.66
SIFT + SVM	0.83	0.34	0.34	0.32
Resnet-50	0.93	0.82	0.75	0.77
Mobilenet V2	0.94	0.89	0.76	0.79
Custom CNN	0.94	0.79	0.83	0.81

Table 5: Weighted averages classification report

	Accuracy	Precision	Recall	F1
HOG + SVM	0.91	.91	.91	.90
SIFT + SVM	0.83	0.74	0.83	0.78
Resnet-50	0.93	0.93	0.93	0.93
Mobilenet V2	0.94	0.95	0.95	0.94
Custom CNN	0.94	0.95	0.94	0.94

Figure 1: HOG + SVM qualitative results

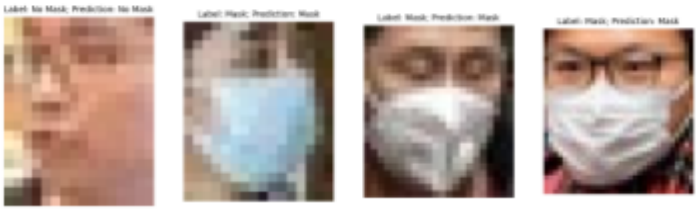


Figure 2: SIFT + HOG qualitative results

SIFT + SVM is also a classic computer vision model approach. However, this model showed worse results than HOG + SVM. The results in the tables show that the model struggled in identifying both true positives and false positives correctly. This could be attributed to its inability to extract the relevant features with SIFT also due to data complexity.

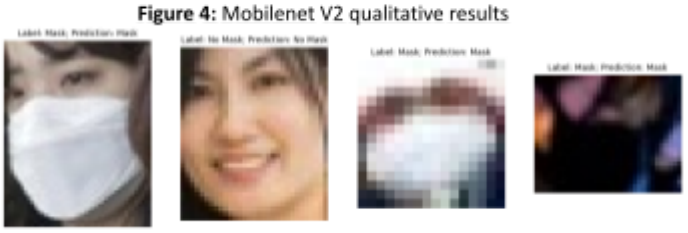
Resnet-50 is a 50-layer CNN model, pretrained on ImageNet. Using transfer learning, Resnet-50 can be tuned for specific tasks such as mask detection. In this case, it showed relatively good results. Mobilenet V2 is a 53-layer CNN model that was also pretrained on ImageNet. Again, using transfer learning, Mobilenet can be adjusted to the task at hand. The model showed relatively good results, and similar results to Resnet-50. For both pretrained models, the backbone layers were frozen, and 2 new linear layers, dropout layers, ReLU and Softmax activation functions were added. Although the architectures were similar, Mobilenet showed slightly better performance. This could be attributed to mobilenet possibly being pretrained on more relevant features to the task than Resnet-50, and the randomly initialised weights for the new added layers. In addition, Mobilenet is more lightweight, so is better for computational efficiency.

Figure 3: Resnet-50 qualitative results

The custom CNN in Keras has 3 convolution layers in addition to batch normalisation, dropout, fully connected,

and pooling layers. This model yielded similar results to Mobilenet. The number of convolution layers are able to extract and capture the features from the images, while the pooling layers reduce the spatial dimensions but also maintain relevant features. This helps prevent overfitting and maximise computational resources. Then, the dropout layers further helps in preventing overfitting, and the batch normalisation layers after each convolutional layer normalises the inputs, reducing internal covariate shift. The fully connected layers increase the model complexity to better capture complex data. Lastly, ReLU and Softmax activation functions are utilised in the model. All of these components of the model contribute to the relatively good performance results [2].

Figure 4: Mobilenet V2 qualitative results



In conclusion, CNNs are generally better at image classification than traditional computer vision models like HOG + SVM and SIFT + SVM. CNNs learn features in a hierarchical fashion, learning low-level features in the initial layers and then learning higher-level features in the following layers [2]. Then, the other layers such as dropout, batch normalisation, pooling, and dense layers further add to the model complexity and its ability to learn complex data. Although the CNNs had significantly better results, the training speeds are much slower because they use deep learning, while HOG + SVM and SIFT + SVM use traditional computer vision methods. However, all 5 models struggled in identifying class 2 (mask improper) images, as shown in Table 6, although the 3 CNN models showed much better scores for class 2 than the SVM models.

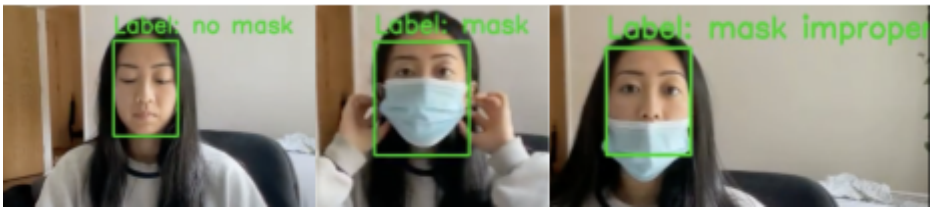
Figure 5: Custom CNN qualitative results

Table 6: Class 2 scores

	Precision	Recall	F1
HOG + SVM	1	0.21	0.35
SIFT + SVM	0	0	0
Resnet-50	0.67	0.42	0.52
Mobilenet V2	0.88	0.37	0.52
Custom CNN	0.62	0.53	0.57

I used the Resnet-50 model on the video, as this showed the best results for the video. It predicted “no mask” and “mask” correctly for almost all the frames, but “mask improper” was only predicted correctly for about half of the frames.

Figure 6: Video screenshots after applying Resnet-50



References

[1] “Face Mask Detection (CNN, ResNet50),” kaggle.com. Available: https://www.kaggle.com/code/gulgaishatemerbekova/face-mask-detection-cnn-resnet50.

[2] J. Maurício, I. Domingues, and J. Bernardino, “Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review,” Applied Sciences, vol. 13, no. 9, p. 5521, Jan. 2023, doi: https://doi.org/10.3390/app13095521. Available: https://www.mdpi.com/2076-3417/13/9/5521.

[3] S. Sadeddin, “Face mask detection trained on ResNet - 50,” Wolfram Cloud. Available: https://www.wolframcloud.com/objects/notebookarchive/submissions/2020-12-6z8vw5z/2020-12-6z8vw5z.nb

[4] Y. Zhang, M. Effati, A. H. Tan, and G. Nejat, “Robust Face Mask Detection by a Socially Assistive Robot Using Deep Learning,” Computers, vol. 13, no. 1, p. 7, Jan. 2024, doi: https://doi.org/10.3390/computers13010007. Available: https://www.mdpi.com/2073-431X/13/1/7.