IN3060/INM460 Computer Vision Coursework report

- **Student name, ID, and cohort:** Morteza Layegh Mirhosseini (220003166) - PG
- **Google Drive folder:** https://drive.google.com/drive/folders/186KAP6KCPt_eUgvUg4PdhdTTIdGwzJ7k?usp=share_link

Data

The data used in this study has a total of 2394 training images and 458 testing images of human faces. Each image is accompanied by a corresponding label, denoted as an integer ranging from 0 to 2, where 0 signifies the absence of a mask, 1 indicates the presence of a properly worn mask, and 2 denotes an improperly worn mask.

Upon examination of the dataset, it was observed that the image sizes are relatively small, with an average height and width of approximately 40 pixels. Furthermore, the dataset exhibits an imbalance in class distribution, with a majority of images belonging to class 1 (images with properly worn masks) accounting for 81% of the dataset, while classes 0 and 2 constitute only 16% and 3% of the dataset respectively.

The video employed in this study to evaluate our model was captured using an iPhone camera. It was saved as an mp4 file and has a duration of 45 seconds, depicting a person wearing a mask correctly, improperly, and without a mask for approximately 15 seconds each. The video has a total of 1309 frames.

Implemented methods

In this study, we have employed four image classification pipelines. The subsequent sections of this study will explain the pre-processing techniques, training approaches, and hyperparameter tuning procedures employed in each of the models.

HOG + SVM

A Support Vector Machine (SVM) model was implemented with Histogram of Oriented Gradients (HOG) as the feature descriptor. Images were resized to 32x32 pixels as a pre-processing step. Augmentation techniques such as random rotation (-20 to 20 degrees) and random horizontal flips were used to increase the number of samples for minority classes due to dataset class imbalance. K-fold cross-validation was used during training for hyperparameter tuning, including different kernel functions and penalty parameter values (C) for the error term.

MLP + HOG

We additionally implemented a multi-layer perceptron with a HOG feature descriptor. The same pre-processing steps were employed as those used for the SVM model. The model was tuned using a holdout dataset, which constituted 10% of the training dataset. We conducted experiments with different learning rates, numbers of hidden layers and neurons, and observed the validation loss to identify the optimal parameters. To prevent overfitting, we used an early stopping call-back.

CNN (Inspired by LeNet) & CNN(Transfer Learning)

Two CNN models were implemented: a basic model and a fine-tuned model using ResNet-18 [1] architecture. The basic CNN model was designed to replicate the architecture of LeNet [2], with image resizing to 32x32, normalization using the mean and standard deviation of pixel intensities for each RGB channel, and random horizontal flip augmentation during training. A validation set was created by randomly separating 25% of the training set for validation to assess model performance during training.

The same preprocessing and validation approach was used for the second CNN model. However, in this case, we leveraged transfer learning by fine-tuning a pre-trained CNN model. Specifically, we downloaded the ResNet-18 [1] model and replaced its final fully connected layer with 3 neurons to match the class count of our dataset. We then trained this modified model using our own dataset.

IMPLENTATION OF MASK DETECTION IN VIDEO

The MTCNN (Multi-task Cascaded Convolutional Networks) pre-trained model [3] is used to detect faces in random video frames. The detected faces are pre-processed and passed as input to our best trained model. The predicted class labels are obtained from the model and the frames with face detection and predicted class labels are displayed next section.

Results

In this section, we will explain the selected hyperparameters after the validation phase and undertake a qualitative and quantitative comparison of the outcomes obtained from various models.

Regarding the HOG+ SVM model, we first choose the HOG parameters based on several experiments and visualization of HOG feature descriptors in the validation phase. We have chosen 9 as the number of orientation bins in the histogram and (4x4) as the size of the cell in which the gradient orientations are computed. Additionally, we have chosen (2x2) as the size of the block over which local normalization is applied.

Next, using a three-fold cross-validation approach and grid search, we were able to identify the optimal kernel function, which was "rbf", and the best penalty value, which was set at 10 for SVM model. Retraining the model with these parameters on the entire training set yielded a validation accuracy of 92 percent.

The selected parameters for the MLP model during the validation phase included two hidden layers, each consisting of 50 and 100 neurons, respectively. The activation function used was Rectified Linear Unit (ReLU), and the optimization algorithm employed was Stochastic Gradient Descent (SGD), with cross-entropy loss function. The training phase stopped at 46 epochs since there was not a significant change in loss for 10 consecutive epochs (early stopping mechanism). The validation accuracy was observed to be 89%.

The architecture of the CNN model consists of two convolutional layers with batch normalization and max pooling, followed by three fully connected layers. ReLU activation functions were used between each layer. The kernel size was 5x5, and there were 6 and 16 filters in the first and second convolutional layers, respectively. The model was trained over 100 epochs with SGD as the optimizer, cross-entropy loss function, and a learning rate of 0.001. The obtained validation accuracy was 0.94. The fine tuned CNN model used the same optimizer but with a scheduled learning rate reduction every 7 epochs. We stopped training the model after 30 epochs as we did not see any improvement in validation loss and accuracy. The final validation accuracy was 0.95.

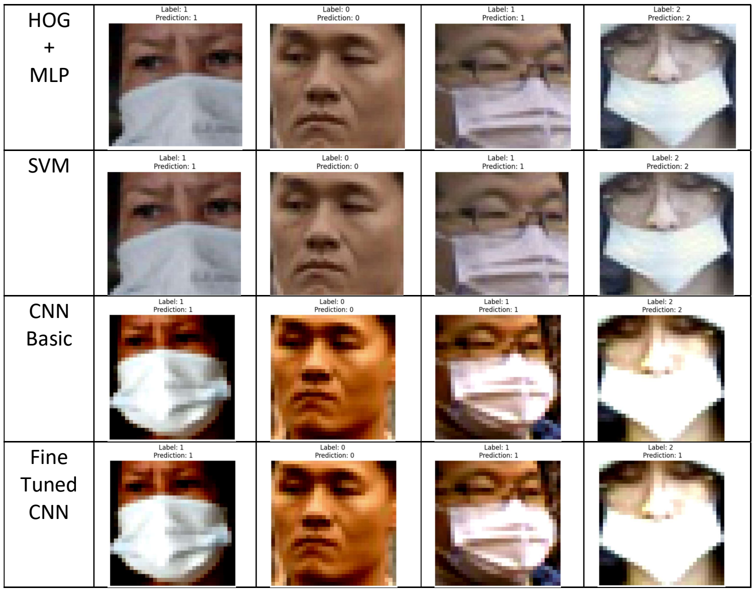


Figure 1. Four examples, labels and predictions for four methods.

The Figure 1 shows four random examples of the predictions made by the models employed in this study. From this random selection taken from the test set, all photos were accurately predicted to the correct label, with the exception of one example where an improper mask (class 2) was misclassified as a worn mask (class 1) by the fine-tuned CNN model (bottom right of the Figure 1).

The Table 1 displays the results of the final models. It is evident that the fine-tuned CNN model exhibits the highest accuracy at 0.95, while the basic CNN model, with only a marginal difference of 1 percent, achieves an accuracy of 0.94. This observation is interesting as the fine-tuned model, employing a more sophisticated architecture such as ResNet18, performs only marginally better than the basic CNN model. Additionally, the HOG+MLP model demonstrates the lowest accuracy at 0.82. It is notable that the HOG+SVM model outperformed the HOG+MLP classifier by a significant margin of 6 percent, achieving an accuracy of 0.88. These findings indicate that an MLP classifier may not be as powerful without convolutional layers compared to its counterparts.

Table 1: The performance of four methods according to accuracy of test, validation, class 0, class 1, class 2 and prediction time.

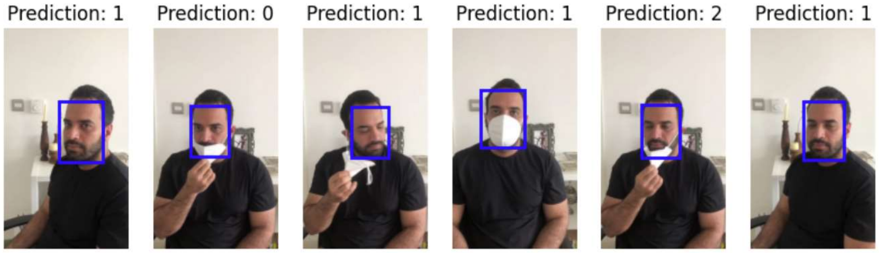
Model	Test accuracy	Validation accuracy	Class 0 accuracy	Class 1 accuracy	Class 2 accuracy	Prediction time (s)
Basic CNN	0.94	0.94	0.75	0.99	0.47	1.79
Fine Tuned CNN	0.95	0.95	0.82	1.00	0.36	5.52
HOG + SVM	0.88	0.92	0.70	0.92	0.36	2.45
HOH + MLP	0.82	0.89	0.58	0.86	0.42	0.02

In terms of class-wise accuracy, the HOG+MLP model performed poorly with an accuracy of 0.58 for class 0 (mask not worn), while the other models achieved higher accuracy above 0.70. The best accuracy of 0.82 was obtained by the fine-tuned CNN model. For class 1 (mask worn properly), all models performed reasonably well with accuracy above 0.85, particularly the basic CNN and fine-tuned CNN models with 99% and 100% accuracy respectively. However, all models showed lower performance for class 2 (mask worn improperly), which was the minority class in the dataset with only 10% of samples. The basic CNN model had the highest accuracy of 0.47 in predicting samples from this class.

Upon analyzing the prediction time for 458 test samples, we found that the finetuned CNN model has the highest prediction time, surpassing 5.5 seconds. This observation can be attributed to the fact that the transfer learning model, ResNet-18, used in this case, possesses a greater number of layers and complexity when compared to the other models, thus justifying the observed difference. On the other hand, the MPL + HOG model stands out as the fastest in terms of prediction time, taking merely 0.02 seconds to generate predictions.

The CNN basic model was selected as our best model due to its high accuracy of 0.99 and its better performance in identifying the minority class (mask worn improperly) compared to other models. Additionally, this model exhibited faster predicting times compared to the fine-tuned CNN model. As a result, we have incorporated this model into our mask detection video function. Several examples of random frames from the test video are displayed in Figure 2, showing that the model predicts class 1 (mask worn) even when the mask is not worn (class 0) in some instances. This observation may arise due to differences in the distribution of video frames compared to the photos used for model training. Furthermore, the presence of facial hair on the person in the video may impact the model's classification accuracy, causing instances of misclassification as "mask worn."

Figure 2: The Basic CNN model’s Predictions of the random video frames



References

1. *MODELS AND PRE-TRAINED WEIGHTS*. Available from: https://pytorch.org/vision/stable/models.html.
2. LeCun, Y., et al., *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.
3. Zhang, K., et al., *Joint face detection and alignment using multitask cascaded convolutional networks.* IEEE signal processing letters, 2016. **23**(10): p. 1499-1503.