# IN3060/INM460 Computer Vision Coursework Report

- **Student name, ID and cohort:** Mohamad Kanso(200007917) - UG
- **Google Drive folder:** https://drive.google.com/drive/folders/1zRYOcP13nh3Aw-smSF3n7LgfVS6LepaK?usp=share_link

**Note:** The test_functions.ipynb was alerted from the version seen in the video showcase as a major error was spotted. def `'extract_hog_features_svm_knn_cnn'` was changed, and a new def `'extract_hog_features_mlp'` was added.

## Data

- The dataset used for this project is a COVID-19 mask detection dataset. It contains images of people with masks, improper masks, and no masks.
- The dataset consists of two folders, one for training and one for testing.
- The training folder contains 2394 images, and the testing folder contains 458 images.
- Each image is of size 224x224x3, where 3 represents the RGB channels.
- The labels of the dataset are mapped as follows: "no_mask" to 0, "mask" to 1, and "improper_mask" to 2.
- The class distribution of the training set is 0: 376 samples, 1: 1940 samples, and 2: 78 samples. The class distribution of the testing set is 0: 51 samples, 1: 388 samples and 2: 19 samples.
- The dataset was loaded and pre-processed using Python and NumPy libraries.
- The pre-processing includes resizing the images to a standard size, converting the labels into numeric values, and splitting the dataset into training and testing sets.
- The dataset was loaded from Google Drive using the PyDrive library.

## Implemented methods.

In this study, we implemented four different classification models to address the given problem. These models were selected based on their effectiveness in handling image classification tasks and their ability to provide a diverse set of techniques for comparison. The implemented models are as follows:

1. Support Vector Machine (SVM) [1]
2. Multilayer Perceptron (MLP) [2]
3. K-Nearest Neighbors (KNN) [3]
4. Convolutional Neural Network (CNN) [4]

**1. Support Vector Machine (SVM)**

The SVM classifier was chosen due to its robustness in handling high-dimensional data and its ability to generalise well to unseen data. To make the SVM model more suitable for image classification, we extracted the Histogram of Oriented Gradients (HOG) features from the images. This process involved dividing the images into cells, computing gradient histograms for each cell, and normalising the histograms in larger blocks. The HOG features were extracted separately for training and test data. The model was trained using the RBF kernel, and the best hyperparameters (C and gamma) were selected using GridSearchCV.

**2. Multilayer Perceptron (MLP)**

The MLP classifier was implemented as a popular and versatile alternative to CNNs for image classification tasks. HOG features were extracted from the images to prepare the data, and the features were standardised using a StandardScaler. A grid search was performed using GridSearchCV to find the best hyperparameters, including hidden layer sizes, the regularisation parameter alpha, and the learning rate schedule. Finally, the MLP model was trained on the HOG features of the training data with early stopping, and the performance was evaluated on the test data.

**3. K-Nearest Neighbors (KNN)**

The KNN classifier was implemented due to its simplicity, non-parametric nature, and effectiveness in image classification tasks. Like the SVM model, HOG features were extracted from the images for the training and test sets. The best value for the number of neighbours (k) was determined using GridSearchCV, which performed cross-validated model training with various k values. The classifier was then trained on the HOG features of the training data using the optimal k value and evaluated on the test data.

**4. Convolutional Neural Network (CNN)**

CNNs were chosen for their proven performance in image classification tasks and their ability to learn hierarchical features directly from the images. The images were first normalised by dividing pixel values by 255, and the labels were one-hot encoded. The CNN model was designed with a reduced capacity to mitigate the risk of overfitting, consisting of three convolutional layers with ReLU activation functions, followed by max-pooling layers, and ending with a fully connected layer and a softmax activation layer for class probabilities. The model was compiled using the Adam optimiser and categorical cross-entropy loss function. Finally, the model was trained on the normalised images with reduced batch size, and the performance was evaluated on the test data.

All four implemented models leveraged different techniques and characteristics to address the image classification task. The SVM and KNN classifiers relied on HOG features to capture meaningful information from the images. At the same time, the CNN and MLP models utilised different types of neural networks to learn patterns in the data. The hyperparameters for each model were optimised using GridSearchCV, ensuring the best possible performance on the given dataset.

## Results

In this section, we present the performance results of the four implemented models for face mask detection: HOG + SVM, MLP + HOG, kNN + HOG, and CNN. We report on their accuracy, speed, and qualitative examples of the obtained results.

| Model | Time | Accuracy | F1 Score (weighted) | Precision (weighted) | Recall (weighted) | Confusion Matrix |
|---|---|---|---|---|---|---|
| HOG + SVM | 16 minutes | 84.72% | 78.00% | 72.00% | 85.00% | Biased towards class 1 |
| HOG + MLP | 2 minutes | 86.24% | 86.05% | 86.66% | 86.24% | Better distribution among the classes, still biased |
| HOG + KNN | 1 minute | 87.77% | 86.00% | 87.00% | 88.00% | Better distribution among the classes, still biased (Similar to MLP) |
| CNN | 20 minutes | 92.79% | 92.69% | 92.89% | 92.79% | More balanced distribution among the classes |

**Discussion:**

The results show that the CNN model achieves the highest accuracy (93.23%), followed by the HOG + KNN model (87.77%), the HOG + MLP model (86.24%), and the HOG + SVM model (84.72%). The CNN model also has a better precision, recall, and F1-score values compared to the other models.

The HOG + SVM model shows poor performance in identifying class 0 and class 2, as it fails to predict any samples from these classes. On the other hand, HOG + MLP and HOG + KNN show better performance in identifying class 0 and class 2 but still struggle to achieve high precision and recall values for these classes. This might be due to the imbalanced distribution of classes in the dataset, where class 1 has a significantly more significant number of samples compared to the other two classes.

The CNN model outperforms the other models because it can learn more complex features and representations from the images. In contrast, the HOG-based models rely on pre-defined feature extraction techniques. Also, the CNN model can handle the imbalanced dataset better than the other models, as it can learn to discriminate between classes even when the number of samples is not equal.

**Qualitative Results:**

The qualitative results show the model's ability to classify images from the test set correctly. For example, the HOG + SVM model fails to identify class 0 and class 2 images, while the HOG + MLP, HOG + KNN, and CNN models can correctly classify some images from these classes. These qualitative examples also confirm that the CNN model performs better in identifying images from different classes, as it can learn more complex features and representations from the images.

The best-performing model, CNN, accurately detects faces with and without masks. It can also identify cases where masks are worn incorrectly, such as when the mask does not cover the nose. However, it may still struggle with specific challenging scenarios, such as when the face is partially occluded or at an extreme angle.


*Figure 1- Example of CNN Results*

## References

[1] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

[2] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. Nature, 323(6088), 533-536.

[3] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27.

[4] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.