# IN3060/INM460 Computer Vision Coursework report

* **Student name, ID, and cohort:** Mohsin Gul (220049189) - PG
* **Google Drive folder:**

https://drive.google.com/drive/folders/1AitAjeLtDlh\_HW8q3ino5ROnxn62VyH5?usp=share\_link

# DataThe dataset used for Face Covering Detection was highly imbalanced having 3 classes of 0, 1 and 2 that shows if mask is not worn, properly worn, and improperly worn respectively. They were total of 2394 images zipped in train and test folder each consisting of images and labels. Class 0 had 376 images, class 1 had 1940 being the majority class, and there were 78 images of class 2 being the least.

The Face Covering Detection were carried out on my own video in mp4 format. It shows me not wearing the mask wearing the mask and improperly wearing the mask in the video.

# Implemented methods

For the face mask dataset, I applied 3 models such as MLP, SVM and CNN. Feature descriptors used with the two models mainly MLP and SVM were SIFT (Scale-invariant feature transform). There were no feature descriptors used for CNN as throughout the training process, CNN learns to extract relevant features from the data itself.

* Opting SIFT and not HOG (Histogram of Oriented Gradients) was mainly due to the reason that considering a task that requires identification and detection of specific features in an image, SIFT works better.
* In contrast, HOG is best suited where there is task of object detection.
* As our dataset, consists of images of face masks that required describing of specific features of face masks on human face, I considered to proceed with SIFT.
* For the pre-processing, the images were having different shapes, so I converted all the images into one shape as (60x60) hight and width respectively. Moreover, I applied normalization to the resized images.
* A lot of images were distorted (blur and noisy), therefore I used image enhancement technique known as image sharpening that is necessarily used for details enhancement.
* I also considered other image enhancement techniques such as histogram and contrast stretching but the prediction reduced.
* As the dataset is highly imbalanced, f1-score with macro and weighted would be used as an evaluation metrics.
* In the pursuit of finding the optimal architecture, I started with training SVM, used 3 different model based on number of 3 different number of clusters and batch size to compare to an extent that the model can be selected. MLP was also trained and optimized using these steps, but CNN was trained without applying the grid search cv.
* The process followed by giving random number of clusters and batch size, after the feature descriptors are extracted from the images HOCW (Histogram of Code Words) is applied to perform clustering by assigning each SIFT feature descriptor to the closest cluster centroid determined by K-means followed by counting the total number of feature descriptors as well.
* From the 3 models, the model that outperformed compared to the other 2 models, the model was selected to perform Grid search cross validation with initializing different list of random values given to hyper parameters so that the optimal hyper parameters are achieved for ultimate results.
* Hyperparameters such as regularization parameter C (minimizes error by maximizing the margin) and gamma that controls the decision boundary size and position in the kernel was considered.
* Similar steps were followed for the MLP architecture by designing wide architecture with 1 layer and intermediate architecture comprising of 2 layers to evaluate the performance giving 48 and 42, 28 hidden neurons respectively.
* For MLP and CNN hyper parameters such as Learning rate and to avoid overfitting early stopping and batch training was considered to increase robustness in learning and to reduce bias.
* Soft max activation function as an output layer because of multi classification and ReLU activation function on the hidden layers were applied to both CNN and MLP architectures.
* Cross entropy was opted as a loss function that determines the difference between predicted and actual probability distribution of classes, also because it is a classification task.
* The CNN architecture was designed by using 2 convolution layers that takes feature maps to extract features from the image. Max pooling layers were also used that are used necessarily for dimensionality reduction in terms of spatial dimension while preserving the information. Additionally, 3 fully connected layers were used to attain the abstract features.
* FCD was applied in the wild on a video of mine that was in the mp4 format. First step was to apply MTCNN (Multi-Task Cascaded Convolutional Networks) which is an algorithm used for face detection. It used cascading architecture to detect landmark and faces in given image, therefore it created bounding box around the faces for further processing.
* The video frames were then taken one by one and applied transformations i.e., resizing and normalizing as required by the CNN model.
* The coordinates returned the by the function after transformation, CNN model was applied as it yielded highest f1- score.
* The predicted classes were assigned to the bounding boxes with different colours to create a sense of uniqueness.

# Results

* The data is highly imbalanced so to evaluate the models I used f1-score with macro averages (un-weighted mean) as it considers the classes equally without discrimination due to support. Moreover, every class is important for this task therefore, considering minority class as important as majority class in terms of treating all classes equally led me to opt macro average.
* For SVM 3 models were designed for training using same kernel and C parameter but different cluster and batch size. In the figure 1 it is evident that 3rd SVM classifier yielded highest macro average of 37% and yielded highest f1-score at class 2, therefore I selected the 3rd classifier to proceed with the grid search cross validation.

Table

Description automatically generated

* Surprisingly after applying grid search cv in figure 2 and achieving the best hyper parameters, the f1-score for minority classes 1 and 2 was drastically declined to 0.05 for 0 and 0 for 2. Moreover, macro average also reduced, thus I opted 3rd SVM classifier to train our best model.

Table

Description automatically generated

* Same steps were taken when experimenting with MLP architecture in fig 3. Intermediate network with 2 hidden layers yielded better result in terms of 0.22 vs 0.14 f1-score and 0.34 compared to 0.32 for macro average. Intermediate network was selected for the grid search cross validation.

Table

Description automatically generated

* Also, MLP model when trained on optimal hyper parameters showed in fig 4 a significant decrease in 0.11 vs 0.22 f1-score and 0.30 compared to 0.34 for macro average when comparing with intermediate neural network model.

Table

Description automatically generated

* CNN architecture was trained in figure 5 once after hyperparameter tweaking and got the best results among the other two models such as SVM and MLP with SIFT for this task. From the figure 3 it infers that I got higher results of f1 score and macro average for all the 3 classes. Macro Average of 0.80 and even 0 .90 and 0.52 f1-scores for minority classes 0 and 2.

Table

Description automatically generated

* In terms of speed the CNN highest amount of time to train of 14 seconds as compared to 10 seconds of MLP and 0.16 seconds of SVM.
* Evaluating the 3 different models SVM, MLP and CNN. CNN outperformed all the other models. In fig of confusion matrix infers that there is high precision and recall for people are wearing the mask and those who are not, thus making this model optimal.
* CNNs are top notch algorithm when solving real life data when the dataset is nonlinear and here it would be useful for achieving higher sensitivity.

Graphical user interface, chart, application

Description automatically generated

* In fig when applying the CNN model to a video, it works exceptionally well even for class of improperly worn mask as seen in the figure 7.

*A picture containing indoor, wall, person

Description automatically generated*