IN3060/INM460 Computer Vision Coursework report

- **Student name, ID and cohort:** Youssef Ayman Abdelmoamen (190054712) - UG
- **Google Drive folder:** https://drive.google.com/drive/folders/1KXHp1pud6w2VGqCrPtGb9YCKZuftj2vQ?usp=share_link
- This is the template for your report. Note that you cannot change font type/size or the margins of the page. Don't exceed 3 pages. Neither appendices nor extra pages for references are allowed.

Data

A dataset with 2394 training photos and 458 test images of human faces was used in this study. Integers ranging from 0 to 2 are used to label the photos, with 0 designating that no mask is worn, 1 indicating that a mask is worn appropriately, and 2 indicating that a mask is worn incorrectly. Each image's labels are provided in accompanying text files. Moreover, the personal dataset was carefully chosen to create a high-quality test set that includes individuals who are wearing and are not wearing masks, as well as individuals who are wearing masks incorrectly, in various contexts with diverse lighting conditions. For each label, a significant number of photos (7 images) were selected in order to assess the model's performance.

Implemented methods

- ***SVM+HOG model***

For the first model, a support vector machine (SVM) was chosen because its success in a variety of datasets and image recognition applications is one factor in its popularity. For instance, a study by Rifai et al. (2011) indicated that linear SVM performed better than other algorithms on a variety of image identification tasks, such as face detection and handwritten digit recognition. In a different study, linear SVM was employed to detect breast cancer, and Gudigar et al. (2019) discovered that it performed with excellent accuracy when compared to other classification methods.

To categorise the photos into one of the three categories (no mask, mask, or badly worn mask), a Linear Support Vector Machine (SVM) model containing Histogram of Oriented Gradients (HOG) features was first selected. With the settings of nine orientations, eight pixels per cell, and three cells per block. These variables control the size of the cell across which the gradients are computed, the number of orientations in the histogram, and the block size over which normalisation is applied, the HOG features were retrieved from the training and testing images. Additionally, the data (labels and images) was loaded, read in grayscale mode and scaled to 64x64 pixels to ensure that all the images are the same size. Then, using a linear kernel, the SVM model was trained using the retrieved HOG features. In order to assess the model's performance, the confusion matrix, classification report, and accuracy of the model were all computed on the test data (more on this in the results section). The model was then saved using the joblib library.

- ***MLP+HOG model***

The reason behind choosing this type of model is that MLPs may learn high-level characteristics from the raw picture data through a series of non-linear transformations in their hidden layers, which is one of the reasons why they are successful in classifying images. Without the requirement for manually created feature engineering, these features can be automatically learned from the data. As a result, the model may perform more generally and scale more effectively.

For the second model, a multi-layer perceptron (MLP) classifier has been used to categorise photographs of people wearing various forms of face masks in this project. Additionally, the features from the photos were extracted using the Histogram of Oriented Gradients (HOG) feature extraction technique. The training set and testing set were both scaled to 64x64 pixels, read in grayscale and loaded from separate locations. Each image's matching text files containing the labels were read. Using the predetermined settings for the number of orientations, pixels per cell, and cells per block, the HOG features were retrieved from the training and testing images (Similar to the SVM). Moreover, The HOG *characteristics that were taken from the training photos and used to train an MLP classifier. The MLP was trained for a maximum of 500 iterations and has one hidden layer with 100 neurons. Then, using joblib, the trained MLP was stored. Using the test set, the MLP classifier's accuracy was assessed, and predicted labels were created. The performance of the classifier on each class was then displayed in a confusion matrix and classification report. More on the results in the following result section.*

- ***CNN***

Finally, the third model chosen was the convolutional neural network, there are two reasons behind choosing this model specifically. The first reason being that it would provide more points for this coursework and the other reason being that studies have shown that in picture classification tasks, CNNs outperform conventional machine learning algorithms like SVMs (Support Vector Machines). For instance, a study by Krizhevsky et al. (2012) shown that the state-of-the-art performance on the ImageNet dataset, which comprises over 1 million images and 1000 categories, was reached by a deep CNN architecture dubbed AlexNet.

TensorFlow was used to create a Convolutional Neural Network (CNN) model for image classification. A dense layer and a Softmax output layer come after each convolutional and max-pooling layer in the CNN model. The sparse categorical cross-entropy loss function and the Adam optimizer were used to train the model over 10 epochs with a batch of size 32. The data was loaded and the model was saved the same way as the SVM and MLP models. Moreover, the model was evaluated on the test set, an accuracy was reached (which will be displayed in the results section) and the confusion matrix and a classification report were presented showing accuracy, F1, Recall and precision.

Results

It is observed from the results that the CNN model performs better in terms of test accuracy than the SVM and MLP models. The accuracy of the CNN is 0.8908, whereas that of the SVM and MLP is 0.8537 and 0.8581, respectively. The CNN has the fewest misclassified samples in each class, according to the confusion matrices, we can see. For instance, the CNN incorrectly classifies just 23 samples in class 0, as opposed to 20 and 19 examples incorrectly classified by the SVM and MLP, respectively. Again, the CNN consistently beats the SVM and MLP models in terms of precision, recall, and f1-score. The CNN, for instance, achieves precision, recall, and f1-score values of 0.91, 0.97, and 0.94, respectively, in class 1, whereas the SVM and MLP achieve lesser values of 0.91, 0.91, and 0.91, and 0.92, 0.91, and 0.92, respectively. However, class 2 is where the CNN falls short, with the lowest precision, recall, and f1-score values among the three models. In this class, the MLP achieves the highest recall and f1-score values, while the SVM achieves the highest accuracy value.

The macro and weighted averages of each model's accuracy, recall, and f1-score are also available for review. The weighted average takes the amount of samples in each class into consideration, whereas the macro average calculates the metrics for each class separately before averaging them. In terms of macro average, the MLP comes in second with a f1-score of 0.64, and the SVM comes in third with a score of 0.62. The MLP is the one with the lowest macro average precision (0.64), followed by the SVM (0.64), and the CNN (0.72). A macro average recall of 0.58 for the CNN, 0.61 for the SVM, and 0.64 for the MLP, on the other hand, is the highest.

The CNN receives the greatest f1-score for weighted average (0.88), followed by the MLP (0.86), and the SVM (0.85). Similarly, the MLP comes in second with 0.86, followed by the SVM with 0.85, and the CNN with 0.88, which is the highest weighted average precision. In terms of weighted average recall, the CNN earns the greatest score of 0.89, followed by the SVM with 0.85 and the MLP with 0.86. Overall, the CNN model performs best across most criteria and has the highest accuracy. The SVM and MLP models still obtain comparatively high accuracies and perform admirably in several metrics, particularly in class 2. This is significant to notice. The particular objectives and specifications of the task at hand ultimately determine which model should be used.

NOTE: the results obtained were through evaluating in each model after training it, the mask_detection function was indeed built but displays multiple errors and I was not able to fix these errors.

References

- Rifai, S., Vincent, P., Muller, X., Glorot, X., & Bengio, Y. (2011). Contractive auto-encoders: Explicit invariance during feature extraction. Proceedings of the 28th

International Conference on Machine Learning, 833-840.

- Gudigar, A., Chokkadi, S., & Raghavendra, U. (2019). Breast Cancer Detection Using Histogram of Oriented Gradients Features and Linear SVM Classifier. Journal of

Medical Systems, 43(1), 5. doi: 10.1007/s10916-018-1124-7

- *Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).*