MGTA611 – Deep Learning Assignment Report

Prepared by Farzan – 235819710

Table of Contents

Introduction:

While the ideas of toying with facial recognition models and convincing people to cover their faces if they are sick is not something new, the rise and impact of COVID-19 pushed both factors into the limelight to reduce the probability of people getting infected by the virus, especially in public places.

To curb raspatory viruses, numerous places mandated a rule of only letting masked people access certain premises to keep the number of infected people at bay, but there was a constant challenge in ensuring that such a rule is followed. Such civil issues led to people experimenting with facial recognition models and expanding the scope to utilize such models to detect if an individual is wearing a mask.

The oldest documented model (Arjya Das Department of Information Technology, Ansari, & Basak, 10-13 December 2020) that uses and references a Machine Learning Model based on packages such as TensorFlow, Keras and OpenCV dates to December 2020. This references a conference that itself cites a project that uses Facial Mask Detection using Semantic Segmentation that was published in 2019 as a basis for their workings. One of the reasons that I found this paper interesting was due to their approach towards rendering images; while CNN 2D layers were used, the authors went an extra step to convert images in their dataset to grayscale.

Business Implications of the Project

One of the major reasons that I chose to work on face mask detection for my Deep Learning project is the level of versatility that this project offers. While the use-case scenario in this case may be restricted to a certain cause, the scope can easily be modified to fulfill a different purpose.

For example, instead of facemasks, the models can be trained to detect whether someone is equipped with Personal Protective Equipment (PPE) or not – a case especially useful in industries where workers may be exposed to hazardous material. Such cases can also be paired up with other security measures to promote a safer workspace.

Data Set and Methodology

The dataset for this project is obtained from Kaggle. It comprises of nearly 7,500 images that are divided between faces of people wearing a mask and faces without a mask. This was the only publicly available dataset that consisted of copyright-free images, so I decided to use this as a base for my model.

To ensure that the dataset is fit for use, all images were converted to 128x128 pixels and in RGB format followed by a conversion of image lists to numpy arrays. With the help of arrays and appropriate scaling, I completed the groundwork for the complex and simple models that I had chosen.

Complex Models

Sequential CNN

The first complex model that was tested consisted of a Convolution Neural Network (CNN) that uses Tensorflow and Keras as a base for its working. The goal was to create a sequential CNN model with multiple layers on top which consist of:

- Conv2D Layers with 32 filters and a 3x3 kernel size.

- MaxPooling2D layers to reduce spatial dimensions of the Conv2D layers.
- A flatten layer to convert 2D layers to 1D layers.
- Dense layers that are fully connected to perform classification.
- Dropout layers to help prevent overfitting.

The training process consists of a 10% validation split and 5 epochs after which the training and performance of the model is evaluated on a different test set. The model hereby achieves a training accuracy of 93%, validation accuracy of 91% and test evaluation accuracy of 90%.

Keras/Pretrained Transfer Model Learning using Mobilenet

Since my dataset can be classified as a binary dataset (a person is either wearing or not wearing a mask – there is no middle ground), I decided to use Mobilenet for a binary pretrained CNN. Base layers were frozen, and the following layers were added to the model:

- GlobalAveragePooling2D to reduce feature map to a single value and make the model slightly less complicated, and
- Two dense layers. One with 1024 neurons, the other with 2 neurons to help keep the values binary.

Data augmentation was used as a preprocessing tool to artificially increase the size of the dataset and data generators were used to train the model on 3 epochs. Evaluation of the model takes place by loading and preprocessing a single image to demonstrate the practical application of the model.

As a result, the model achieved high accuracy and showcases how relevant features from an image can be extracted towards effective usage for a binary task. The approach also highlights the potential need for large training datasets to prevent overfitting as well as how pretrained models can speed up the overall processing.

Simple Learning Models.

Simple Sequential Model using MNIST

This was a rather basic approach towards the model that only consists of a Flatten layer to convert 2D images followed by a dense layer for non-linear transformations. This was done to keep the outcome binary even though MNIST models are generally multi-class classifications.

As a result, the training and testing logs with 10 epochs and a validation split of 20% demonstrated unusual negative losses with a decreasing trend, which indicates that this is not a suitable approach for this model since loss values typical point towards zero. However, a second attempt corrected this, but the approach remains ambiguous. While this may count as a practical demonstration of MNIST, the peculiarities in the result make it unsuitable for further analysis.

Support Vector Machine (SVM) Approach

Once again, I decided to treat this as a binary classification model using SVM and avoiding the use of Convolutional Neural Networks altogether to keep this approach simple. The preparation stage consists of images being converted to RGB and 128x128 pixels along with assigning labels for with/without mask.

A Histogram of Oriented Gradients (HOG) feature has been used to capture edge and structure of gradients (Kh Tohidul Islam, 2017) and generally works well in terms of content detection. This was followed by a train-test split with 20% of the data reserved for testing. As mentioned in this paper, image processing that used a combination of Artificial Neural Networks (ANNs) and HOG features achieved accuracy levels as high as 99.0% in practical applications.

With the help of feature scaling and SVM training with a linear kernel (since I opted for a binary approach), the trained model is evaluated by its ability to predict the label (either 0 for without mask, or 1 with mask), and accuracy is measured with the help of precision, recall, and F1-score for each class. The model and approach resulted in an overall accuracy of approximately 83%, strongly showcasing that simple models can be just as effective towards such problems.

Comparative Analysis between the Methodologies

The comparison between CNN and SVM highlights several fascinating insights. As demonstrated in the code in the technical appendix of this report, CNN models rely on their deep learning capabilities and perform well in terms of directly learning from raw images. Such models can highlight complex patterns and select relevant features without manual feature extraction. However, this comes at the cost of large amounts of computing power as well as the need for labeled data for training.

On the other hand, SVM models perform their best when a careful, structured approach is taken (such as in the case of an HOG-based approach). Feature selection is of paramount importance, as the case with traditional Machine Learning approaches. While a greater degree of preprocessing and domain knowledge is required, they still end up taking less computational power to execute.

Conclusion

To conclude, we can use our approaches towards each type of model to demonstrate identical accuracy figures regardless of the approach. CNNs perform their best when computational resources are not limited, whereas SVMs with the right feature engineering manage to catch pace with less overheads. Keeping such factors aside, it remains imperative to observe broader themes, such as the impact of trade-offs in model complexity, approaches taken towards preprocessing data, and the potential impact of feature scaling with complex models.

References

Arjya Das Department of Information Technology, J. U., Ansari, M. W., & Basak, R. (10-13 December 2020). Covid-19 Face Mask Detection Using TensorFlow, Keras and OpenCV. New Delhi, India: IEEE.

Kh Tohidul Islam, R. G.-M. (2017). Performance of SVM, CNN, and ANN with BoW, HOG, and Image Pixels in Face Recognition. *International Conference on Electrical & Electronic Engineering (ICEEE)*. Rajshahi, Bangladesh: IEEE.

Technical Appendix