ECE 4424 - Final Report

1 Authors

Janak Majeethia
Akhil Bhasin
Preethi Ramamurthy

Abstract

Machine learning is being applied all around us to solve scientific and societal issues. Since the rise of COVID-19, the use of face masks has been encouraged in order to reduce the number of cases. Our project focused on detecting whether or not individuals were wearing face masks. It is different from existing works because it not only studied the detection of a face mask being worn, but also if it was worn correctly and not under their nose.

Our intention is to survey exactly how face-detection works – where machine learning comes into the process vs. computer vision techniques – and study what makes a good object detection model and what hyperparameters of a neural network with the above task are influential towards the performance of the model. The study for this topic will include exploring the chosen data, and referenced code in order to provide visuals and explain certain features of such a model.

2 Statement of the Problem

Today, the world is experiencing a crisis, and it is important that humans work together to overcome the situation we have put ourselves into. In order to help the efforts in improving response to COVID-19, our group was keen on exploring software and machine learning applications that help make society safer.

Our goal in trying to contribute to this effort is to explore, analyze and learn what goes into the art of object recognition with the culmination of computer vision concepts and machine learning/deep learning algorithms.

As we know, there is a lot of policy and law enforcing the use of face masks in the country today. Although it is considerably hard to enforce such things, having a computer aided AI software that would respond to visual input, would really assist agencies and businesses in ensuring the safety of their customers and the local society. There are already certain detector algorithms that will detect whether a person is wearing a face mask or not, however, we wanted to explore a “smarter” version of this implementation and analyze whether a detecting model using a CNN can go further and identify if a person is wearing a face mask – but incorrectly.

We started with attempting to reproduce results from a given implementation, we quickly realized that this implementation required a great deal of computing power. With the given data, however, we were able to explore how an implementation could work to satisfy the solution to the mentioned problem.

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

3 Analysis/ Results

In this section we will talk about what we discovered as we dove into the training data and the model and seeing how the referenced implementation performed. Please note that the following discussion is based on the existing implementation found online.

As the team began looking into the facemask-detection-dataset we noticed that the sizes for the input images vary throughout the set. We also noticed that the train dataset contains 4 fundamental cartesian points as part of the data point in the frame. These 4 points represent the rectangular box that borders the face of the person in each picture. This fact guarantees that the model we will fit will have to be fed images that contain a face and the box coordinates of the face in that picture. For this reason, our team has also found it time consuming to add our custom data points into the dataframe to add customization to our project as this involves calculation for each image.

Additionally, the purpose of having these box coordinates was to achieve successful results from the ReL activation unit which performed the first convolution of the input image. Instead of convolving the entire input image, the first layer in our CNN was given a snippet of the face in the image. We understand that this limited the variety of inputs that can be fed to our model, however, this approach ensured lower computational load on our devices, which in turn allowed us to tweak our model more often for a better learning experience.

In this implentation MTCNN was used to automate the process of creating the bounding boxes, but Computer vision techniques from OpenCV proved to be extremely beneficial due to its robust number of methods as well.

The first was the use of OpenCV's built-in person-detection using HOG(Histogram of Oriented Gradients) + Linear SVM. This automated the process of creating the bounding boxes by isolated people in a picture. Essentially doing the same as an MTCNN

Another implementation that increased accuracy was the use of non-maxima suppression. This is useful when dealing with multiple, overlapping bounding boxes. This helped reduce the number of false-positives reported by the detector and also helped when several faces were in a picture.

Initially, the images were only being read to detect if a face mask was worn or not. To increase accuracy of whether or not the mask was worn properly or not, we created another program which tested the effect of changing the box coordinates in the input data. The model first used the box as the start to the convolutional process, after which the coordinates were mapped to OpenCV rectangles for annotation of faces in images. We then implemented the changes back into our original program and the accuracy increased exponentially. The outputs that were obtained were classified correctly into the four categories.

4 Graphs and Findings

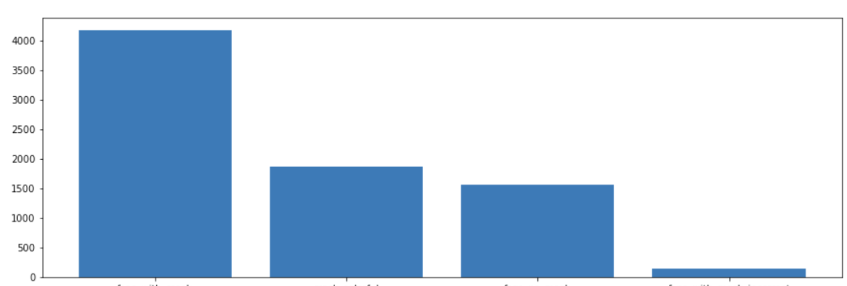

Figure 1: The histogram below shows the most common features from the set and their counts in the train split.

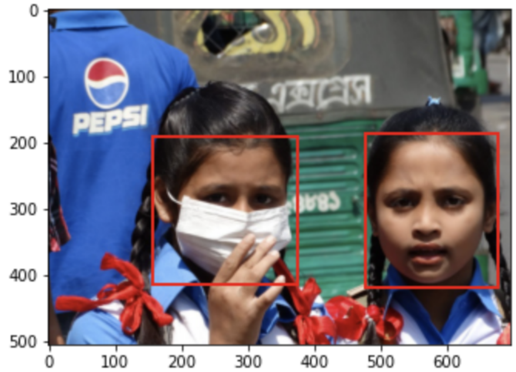

Figure 2: The following image plot shows the rectangular box coordinates that are contained in the train data, and how the coordinates are represented visually.

	name	x1	x2	y1	y2	classname
0	2756.png	69	126	294	392	face_with_mask
1	2756.png	505	10	723	283	face_with_mask
2	2756.png	75	252	264	390	mask_colorful
3	2756.png	521	136	711	277	mask_colorful
4	6098.jpg	360	85	728	653	face_no_mask

Figure 3: Showing an example snippet of a CSV file that contains the bounding box data for input images

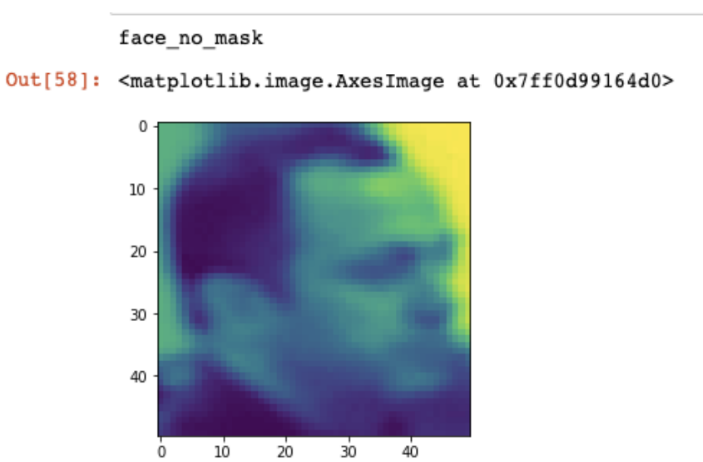

Figure4: An example of the facial snippet from the boxes of each image, along with the label for

5 Analysis of Graphs

Figure 1 indicates the frequency of some features contained in our dataset out of 6024 images. This figure shows the low number of data points that are labeled as `'face_with_mask_incorrect'`. Initially, we wanted to identify various incorrect usages of masks in particular, however, this graph helps us understand that our training data is not strong in that aspect. Thus, as discussed later in the conclusion, we will think about redirecting our efforts.

Figure 2 is a good representation of usefulness of the box coordinates in our input data. The image is seen to have two faces, the model will first use these boxes as the start to the convolutional process, after which the coordinates will be mapped to OpenCV rectangles for annotation of faces in images. This will allow for multiple faces in input images and a unique classification for each face in the image.

Figure 3 shows how our original bounding box data is sourced and formatted. This table shows that even for the same picture, there can be multiple bounding boxes with different labels, as there can be multiple people in one image, out of which all should be classified by our model.

Figure 4 is a direct print of 2D list variable 'data' that we will use to convolve with keras layers. In this specific example, the 86th image (data[85][0]) is plotted, and right before it, the appropriate label `'face_no_mask'` (data[85][1]) is printed. The figure shows a successfully resized snippet of a face, and its training label.

6 Interpretation of results

Approaching the end of this report, we will now discuss some larger findings with respect to the process of object recognition and classification through input images. Additionally, our chosen implementation has been marked to perform with 98.52 percent accuracy and a 4 percent categorical cross entropy loss.

As it was mentioned in the project proposal, the data was gathered and studied in order to develop a program that would assist in detecting face masks on human faces and whether or not these masks were worn correctly. We were able to gather our train data which fit properly into our project approach.

The data set we gathered contained many images of training images, however, as shown earlier in the report, there was a lack of training images where the mask was worn incorrectly. This made us question how the performance was influenced by the number of this data.

The larger number of features than expected made the dataset more applicable to real-life situations since the sizes of the masks and the people wearing them are not definitive. Once the images were put into the program, it was seen that `'face_with_mask'` made up the majority of the data set while the `'face_with_mask_incorrect'` was very low in comparison. Due to this, our team discovered that perhaps the accuracy stayed high because even the test data did not have enough examples of incorrect mask usage. In a real application of this software there would be a lot of cases of incorrect usage, and the model in that case would not work as well, thus, more training data labelled incorrect mask usage would be needed.

7 Contributions

Janak Majeethia: Initial Findings, Graphs and Findings
Akhil Bhasin: Analysis of Graphs
Preethi Ramamurthy: Statement of the problem, Abstract, Conclusion