Face mask detection

Andrej Betik (456604)

25.5. 2022

1 Introduction

The aim of the project was to train a classification model which would be able to detect whether a person is or is not wearing a mask. The model could be used for further development, e.g.: monitoring high-risk virus transmission areas with a video camera and subsequent determination of the rules obeying.

2 Dataset

In order to train and evaluate a model we needed lots of images with people wearing masks. There is annotated face mask detection dataset publicly available on Kaggle [1]. Dataset contains 853 images belonging to the 3 classes, as well as their bounding boxes. The classes are: with mask, without mask, mask worn incorrectly. The distribution of the classes and an example of the image and its corresponding annotation file are shown in the following figures.

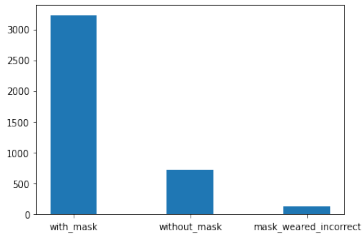


Figure 1: Dataset classes distribution

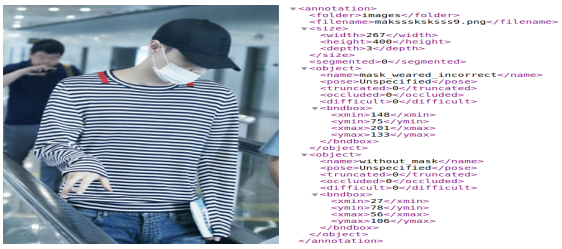


Figure 2: Sample image (png) and its corresponding annotation file (xml)

3 Existing solutions

Generally, most of the publication focus is on face construction and identity recognition when wearing face masks.

Mohamed Loey [2] worked on a a hybrid model using deep and classical machine learning for face mask detection. The proposed model consists of two components. The first component is designed for feature extraction using Resnet50. While the second component is designed for the classification process of face masks using decision trees, Support Vector Machine (SVM), and ensemble algorithm.

In [3], the authors developed a new facemask-wearing condition identification method. They were able to classify three categories of facemask-wearing conditions. The categories are correct facemask-wearing, incorrect facemask-wearing, and no facemask-wearing. The proposed methed has achieved 98.70% accuracy in the face detection phase.

4 Used models

4.1 Haar cascade

Approach based on a cascade function trained from a lot of positive and negative images. It is then used to detect objects in other images. Full documentation can be found at OpenCV [4] website and the original paper [5].

4.2 VGG19

VGG-19 [6] is a convolutional neural network that is 19 layers deep. Pre-trained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

The model was fine-tuned with additional dataset [7] of images of people wearing masks. This dataset contained 6k training and 6k testing samples.

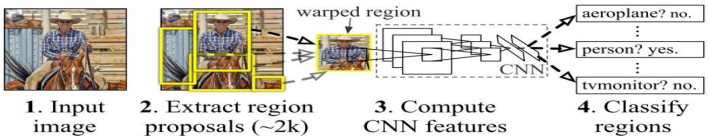
We used this model in a combination with Haar cascade model the following way:

1. Detect face coordinates with Haar Cascade
2. Crop the image
3. Detect whether a person is or is not wearing mask (VGG19 model)

4.3 Region-based convolutional neural network

For object detection, region-based CNN detection methods are now the main paradigm. The main contribution of R-CNN is extracting the features based on a convolutional neural network (CNN).

Model specification and parameters
We used the newest extension of R-CNN model: Faster R-CNN. As the name suggests, the main contribution of this variant is the quicker training. The pre-trained Faster-RCNN model is available in PyTorch. The model was pre-trained



on Common Objects in Context (COCO) dataset. The model was then fine-tuned with images from the Kaggle dataset mentioned in the section Dataset.

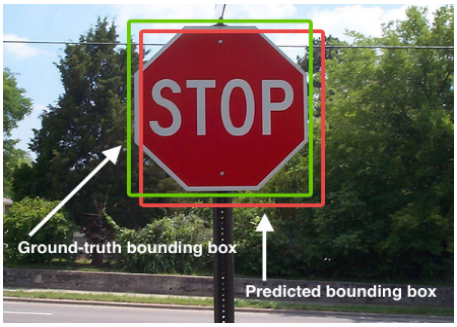
5 Evaluation

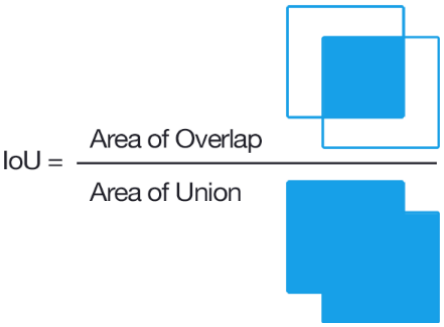
5.1 Issue - Multiple faces in a single image

In case of multiple faces in a single image the order of the model output face coordinates and its labels does not always match the order of the annotation face coordinates and its labels. See the following example.

```
Prediction boxes
[289.57043  97.09577 355.6581  232.76892]
[ 98.6173   89.08514 176.85394 166.1369 ]
[177.82425  52.63085 274.81403 161.35594]
Prediction labels
[2 2 3]
Annotation boxes
[181, 54, 273, 162]
[99, 87, 176, 165]
[289, 99, 355, 233]
Annotation labels
[3, 2, 2]
```

Therefore we used Jaccard distance to find and match the closest annotation box.





5.2 Evaluation metrics

Box overlap: Jaccard distance between prediction and annotation boxes (range [0,1]).

$$\text{Face box coordinate evaluation} = \frac{\text{Sum of overlaps}}{\text{Number of boxes}}$$

Label overlap: prediction label equals annotation label.

$$\text{Labels evaluation} = \frac{\text{Number of overlapping labels}}{\text{Number of labels}}$$

5.3 Results

The following table shows the results of two approaches: Haar cascade with the help of VGG19 and Faster-RCNN. The numbers in the table represents the accuracy of the models. In case of Haar cascade + VGG19 approach, we tested 716 detections from which 145 prediction labels were correct, wheras in the case of Faster-RCNN we tested 219 detections from which 213 were predicted correctly. The number of tested images in the case of Faster-RCNN is lower because we used images for fine-tunning the model. The evaluation of coordinates is in range $< 0, 1 >$ which is the sum of image overlap percentage over the number of images (as mentioned in evaluation metrics).

	Haar cascade + VGG19	Faster-RCNN
Box coordinate evaluation	0.49	0.8
Labels evaluation	0.20	0.97

We can see that Faster-RCNN model performs way better. Faster-RCNN is more robust model and we fine-tunned it with our dataset. Haar cascade was not fine-tuned with our data and is rather light-weight, fast and easy to use model.

The confusion matrix presents Faster-RCNN model performance considering the individual classes prediction.

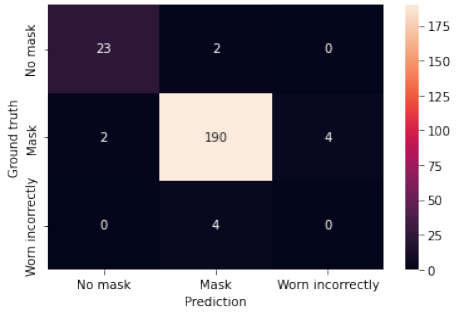


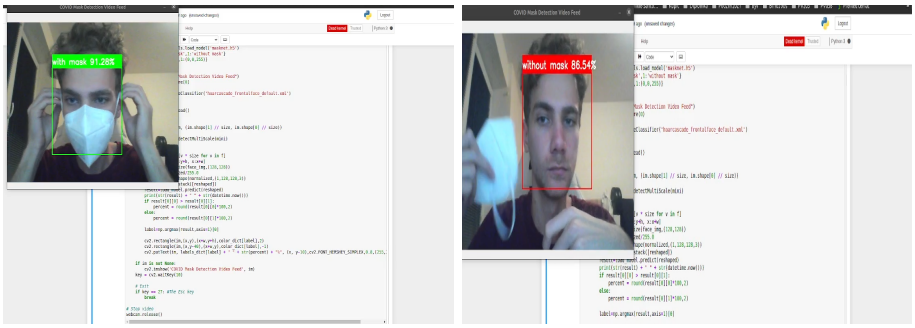
Figure 3: Confusion matrix model Faster-RCNN

5.4 Incorrectly classified images

Since there was a little portion of mask detections classified incorrectly we could take a look at them. Most of them were of the class "Mask worn incorrectly". That might be caused by the small number of training samples (see classes distribution). We could improve our model by oversampling this class.

5.5 Testing from webcam

The goal of this project was to create a model which could be used for further development. A possible use case would be to monitor high-risk virus transmission areas with a video camera and determine if the rules are obeyed or not. Therefore we also tested the model by using webcam. The model successfully recognizes the mask whenever we put it on or off with very high probability.



6 Repository details

The project is available at Github [8] repository. Since both model files are larger than github limit they will be included only in the zip file. Faster-RCNN model was trained and evaluated on CUDA on AWS cloud SageMaker (pytorch kernel).

7 Installation and startup instructions

- Installation and startup library requirements: python, jupyter notebook, python libraries that are being used within the notebooks. We also suggest to use CUDA for training Faster-RCNN model.
- Startup instructions:
  - Download the training data from our repository or from Kaggle and adjust the paths according to your folder structure. There are jupyter notebooks to train each of the models. You can run these notebooks and train the model yourself or just load already trained models from the repository. The models are ready to be used for face mask detection. There are also evaluation notebooks which gives us an insight how well the models perform.

References

[1] Face mask detection dataset. https://www.kaggle.com/datasets/andrewmvd/face-mask-detection. Accessed: 2022-31-5.

[2] Mohamed Loey, Gunasekaran Manogaran, Mohamed Hamed N. Taha, and Nour Eldeen M. Khalifa. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the covid-19 pandemic. *Measurement*, 167:108288, 2021.

[3] Bosheng Qin and Dongxiao Li. Identifying facemask-wearing condition using image super-resolution with classification network to prevent covid-19. *Sensors*, 20(18):5236, 2020.

[4] Haar cascade documentation opencv. https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. Accessed: 2022-31-5.

[5] Paul viola, michael j. jones, "robust real-time face detection",. International Journal of Computer Vision 57(2), 2004. Accessed: 2022-31-5.

[6] Vgg-19. https://www.mathworks.com/help/deeplearning/ref/vgg19.html. Accessed: 2022-31-5.

[7] Face mask detection 12k images dataset. https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset. Accessed: 2022-31-5.

[8] Github repository. https://github.com/xbetik/PA026. Accessed: 2022-31-5.