UNIVERSITÄT BIELEFELD

Technical Report

Application of YOLOv5 for Mask-Type Prediction

T. Lenze Master's student in Data Science
Bielefeld University, Centre for Statistics

27 February 2022

Summary
For the „AIBlitz XIII" challenge, hosted by AIcrowd (Link 1), we applied YOLOv5 to the sub-task of mask prediction. Trained on the provided dataset, it was possible to predict the mask type and its bounding box, in a given image, with an accuracy of 99.3%, which was the top score in this task.

Problem Statement
The sub-task "Mask Prediction" was one of five tasks which the AIBlitz XIII challenge consisted of. The aim of this task was to develop a method that is able to predict the mask type worn by a person in a picture and also to predict the bounding box of the mask (img.1).

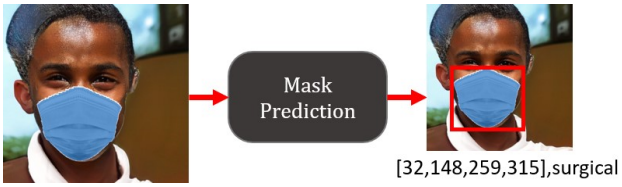

Image 1. Left: The input of the mask prediction, a picture of a person with a mask. Right: The output is a picture with a bounding box, the bounding box coordinates and the mask type.

Dataset
The provided dataset was split into three different sets. Train, validation and test sets. Each set contains 5000, 2000, and 3000 512x512 images respectively. Each image includes one of 4 mask types: Surgical, N95, KN95 and Cloth (img.2). Also provided were CSV-files for the train and validation image sets which contained the information on mask-type for each picture and the corresponding bounding box of the mask in pixel coordinates (img.3).

Image 2. Samples of the data set for each mask type. L.t.r.: N95, cloth, surgical, KN95

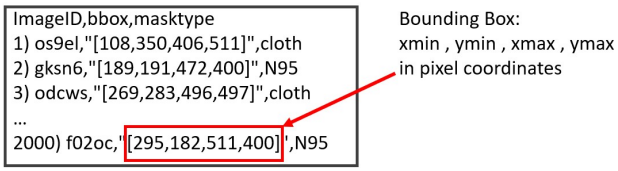

Image 3. Data structure of CSV files: ImageID, bounding box and mask type. Also the format used for the bounding box on the right.

Prediction Method
YOLOv5 (Link 3) was chosen as prediction method. YOLOv5 is the fifth iteration of YOLO (You only look once) , which is a state-of-the-art, real-time object detection system that is also capable to produce bounding boxes around the detected objects.

Setup
YOLO requires a specific format and file hierarchy. For this the following fire structure needed to be implemented in the same directory as the YOLOv5 folder:

/datasets/mask_prediction/images/train/
/datasets/mask_prediction/images/val/
/datasets/mask_prediction/images/test/
contains the training, validation and test images.

/datasets/mask_prediction/labels/train/
/datasets/mask_prediction/labels/val/
contains the training and validation labels.

After setting up the file hierarchy, labels in the YOLO format need to be created and moved to the respective "labels" folders. These labels are .txt files for each image, named the same as their corresponding images. For example the image **.../images/train/abc.jpg** needs to have a counterpart **.../labels/train/abc.txt**. Each label contains the type of object it contains a numeric value (starting with 0) and the bounding box in this format:

<object-class> <x> <y> <width> <height>

Since this format is different than the one in the CSV's these values needed to be converted. Also to note is that YOLO uses values that are relative to image size and are not pixel based. To convert the values from the CSV`s to the YOLO format following formulas were used:

$$x = \frac{\frac{x_{max} - x_{min}}{2} + x_{min}}{ImageSizeInPixel} \qquad width = \frac{x_{max} - x_{min}}{ImageSizeInPixel}$$

$$y = \frac{\frac{y_{max} - y_{min}}{2} + y_{min}}{ImageSizeInPixel} \qquad height = \frac{y_{max} - y_{min}}{ImageSizeInPixel}$$

The last step required for the setup this to create a **mask_prediction.yml** in **yolov5/data** which contains the paths to the training and validation images as well as the number and names of classes (see notebook in Link1).

Training
After the setup the training in YOLOv5 can be initiated by just one command with parameters that indicate to the program the image size in pixel, batch size, number of epochs, the path to the mask_prediction.yml file and the size of YOLOv5 network to train on(see notebook). Several configurations have been investigated. The training time per epoch for the training set of 3000 images and a batch size of 16 was about 15 minutes on a Tesla K80 via Google Colab and about 1:30 minutes on a local RTX3070. The training produces two sets of weights, **last.pt** and **best.pt** which represent the weights after the last training epoch and the one with the best validation score respectively.

Inference
The inference can also be started with a single command and provides multiple optional parameters for the output. Per default the inference produces only the images with a bounding box and a label. To get the bounding-box-coordinates the parameter **--save-txt** was required to generate labels in the YOLO format. These labels, then needed to be converted back to the format in the CSV files to be submitted for scoring. The scoring was based on the Average Precision of mask type and bounding box (@ IoU=0.50:0.50)

Optimizations
All tested training configurations had some images where the default inference was not able to identify a mask. For these images the inference was conducted a second time with an additional parameter that reduce the confidence threshold. With that it was possible to find masks on all images. Other methods, for example color-based image processing, are imaginable, but this approach was ultimately chosen for its simplicity and sufficient results. In addition to the optimization of the inference, the enhancement of the training dataset by adding of the validation dataset was investigated. Due to a then missing validation dataset the only reliable quality metric for this approach was the submission score. The Score was in general, higher with this method, but didn't produce the best score.

Results
After conducting the inference on the test image dataset the results were uploaded and scored. The best result, with an accuracy of 99.3%, was achieved after training for 100 epochs, with a batch size of 16 images and with the YOLOv5l model and choosing the **last.pt**-weights for the Inference. To note is that most other configurations also achieved sores in the high nineties. Some of them are listed below:

Epochs	Batch Size	Score	Inference weights	Network Size
50	16	98.2%	best.pt	YOLOv5m
50	16	97.9%	last.pt	YOLOv5m
200	16	98.8%	best.pt	YOLOv5m
200	16	98.4%	last.pt	YOLOv5m
100	16	99.0%	best.pt	YOLOv5L
100	16	99.3%	last.pt	YOLOv5L

Links
1)Github with image datasets, CSV's & the notebook used for the best sore:
https://github.com/IZMEHD/MaskPrediction
2)Link to the AIBlitz XIII challenge:
https://www.aicrowd.com/challenges/ai-blitz-xiii
3)YOLOv5:
https://github.com/ultralytics/yolov5