

Real time Face Mask detection based on deep learning object detection method

Supervisor: Liang Liang

The purpose of this project:

Due to the outbreak of covid, masks have become one of the indispensable items in people's daily life. Wearing a mask is not only to avoid infection from others, but also to avoid getting others infected. Schools with large number of young people are one of the high incidence areas to get infected. Therefore, the mask should be highly required for every teacher and student before they entered in the classroom. Here I used "You Only Look Once" (YOLO) objection detection model to implement a **real time** face mask detector. Whenever any students or teacher didn' t wear mask when entering the classroom, this model will automatically detect it and tell them to wear mask.

How does YOLO works:

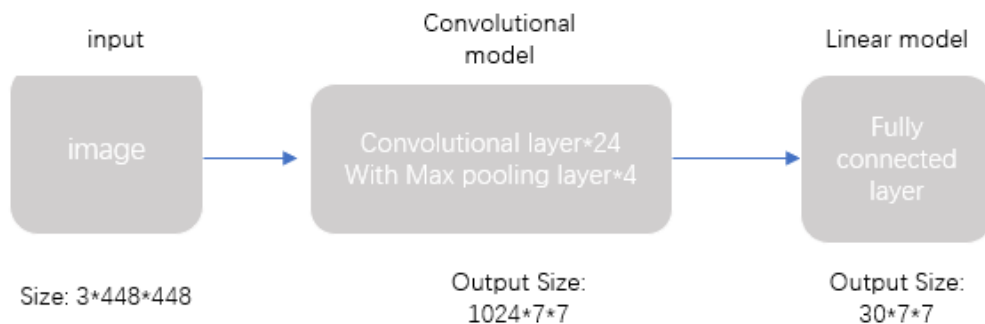


Figure1: the construction of YOLO model

YOLO model is a convolutional model with 24 convolutional layer to extract features and 4 Max pooling layer to decrease the parameters to prevent overfitting and 2 fully connected layers to do prediction. The input of YOLO model is an image with size 3(RGB 3channels)*48*48 and the output is a Tensor with size 30*7*7.

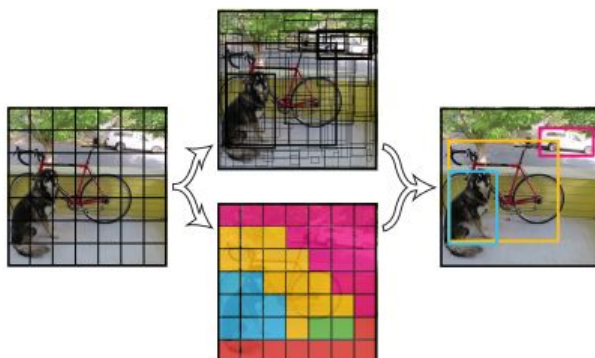


Figure2: How does YOLO works

Where 7*7 means the model divide the input image into 7*7=49 grid cell, each grid cell got 2 bounding boxes. The 30 dimension is composed of 3 parts (5+5+20), where the first two 5

dimension represents the x, y, width, height, and the conditional probability. The last 20 dimensions represents the class probabilities.

Process :

1. Data

- Data set: open sourced face mask dataset collected by Wuhan university: <https://github.com/X-zhangyang/Real-World-Masked-Face-Dataset>.
- With mask on: 10000 samples
- Without mask on: 5000 samples

2. Data preprocessing

- Label those images
- Resize those images into 448*448*3
- Split the data into training, validation, and testing sets by 8:1:1
- Reconstruct the DataSet class based on our data
- Defined DataLoader to load our data

3. Training

- Construct the YOLO model and define the loss function and optimizer
- Set both the model, the optimizer, into GPU mode to train faster.
- Put our training data from Dataloader into this model and train it for 100 epochs
- Based on the loss function to use back propagation and gradient descent to train the parameters

4. validation

- do validation and generate report of accuracy

5. Camera connection and test

- Connect to a camera demo and convert the video into 30 images per second.
- Put those images into our pretrained YOLO model and see the result.

Expected Result:

Ideally when we put our mask on, there would be a box bounding our face with "mask in on" label. When we don't put our mask on, the label would turn into "no mask" .