

NATIONAL UNIVERSITY OF COMPUTER AND

EMERGING SCIENCES, KARACHI



**YOLO RESEARCH PAPER REPORT**



Group Members: Instructor:

Noman Anjum: 17K-3755, Dr. Muhamma Rizwan

Muhammad Ahsan: 17K-3650,

M. Hassan Ahmed: 17K-3654,

Muhammad Mustafa Rafiq: 17K-3795

Department of Computer Science

4 May, 2021

**Abstract:**

The exponential spread of COVID-19 in over 215 countries has led WHO to recommend face masks and gloves for a safe return to school or work. We used artificial intelligence and deep learning algorithms for automatic face masks and gloves detection in public areas. We investigated and assessed the efficiency of popular deep learning algorithms of YOLO (You Only Look Once) for the detection and proper wearing of face masks and gloves trained over a data set of 8250 images imported from the internet. YOLOv3 is implemented using the DarkNet framework. The proposed model have been developed to provide accurate multi-class detection (Mask vs. No-Mask vs. Gloves vs. No-Gloves vs. Improper). When people wear their masks improperly, the method detects them as an improper class. The introduced models provide accuracy of (90.6% for YOLO) for multi-class detection. The systems’ results indicate the efficiency and validity of detecting people who do not wear masks and gloves in public.

**Introduction:**

Object detection is one of the classical problems in computer vision where we work to recognize what and where specifically what objects are inside a given image and also where they are in the image. The problem of object detection is more complex than classification, which also can recognize objects but doesn’t indicate where the object is located in the image. In addition, classification doesn’t work on images containing more than one object. YOLO uses a totally different approach. YOLO is a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. YOLO is popular because it achieves high accuracy while also being able to run in real-time. The algorithm “only looks once” at the image in the sense that it requires only one forward propagation pass through the neural network to make predictions. After non-max suppression (which makes sure the object detection algorithm only detects each object once), it then outputs recognized objects together with the bounding boxes. With YOLO, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This model has a number of benefits over other object detection methods:

* YOLO is extremely fast
* YOLO sees the entire image during training and test time so it implicitly encodes contextual information about classes as well as their appearance.
* YOLO learns generalizable representations of objects so that when trained on natural images and tested on artwork, the algorithm outperforms other top detection methods.

**Experimental Setup:**

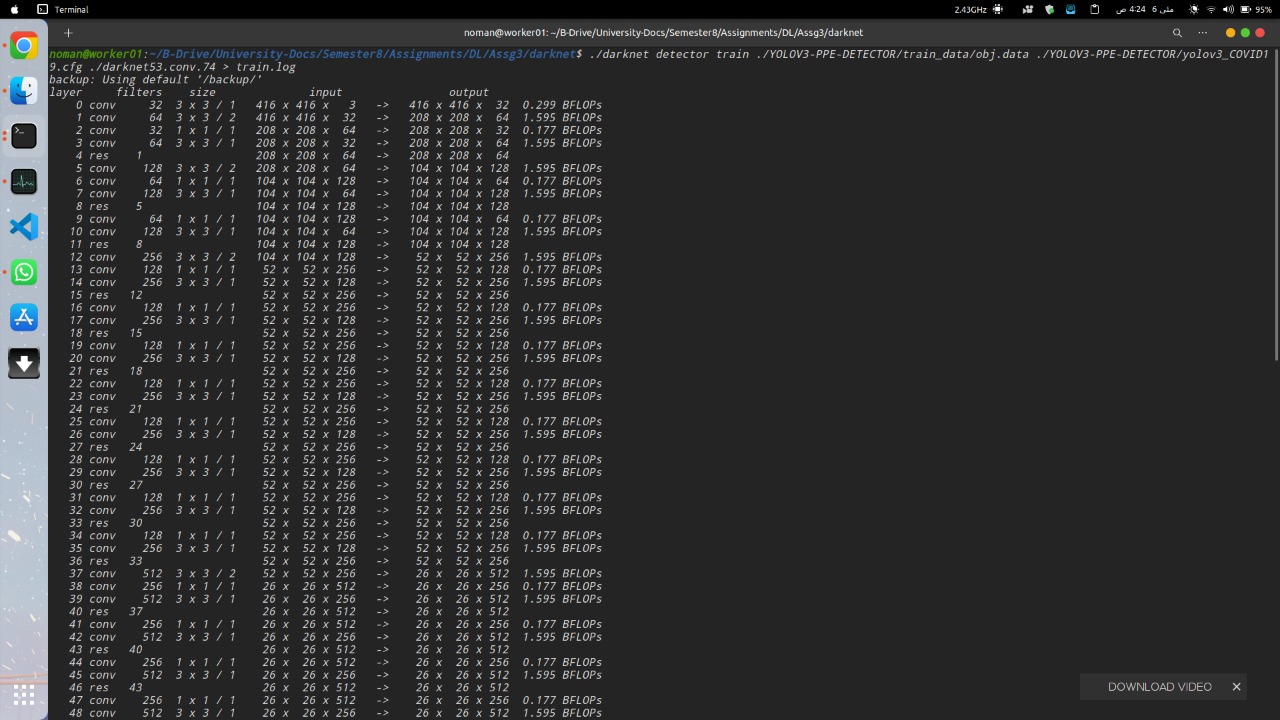
We used provided raw dataset and through the python script we separated the images that had annotation files by traversing the raw data. Name of separated images were put in the train.txt file and 1000 images from separated data were put in test.txt for testing purposes. Obj.names file contains all the possible outputs (classes) that are mask, improper, no-mask, glove, and no-glove. Obj.data file specifies the Number of Classes and path of train.txt, test.txt and obj.names along with the path to save the weight for trained network. Most important file to train YOLO network is yolov3\_COVID19.cfg. This configuration file contains all the parameters that specifies model’s architecture, which are as follows: batch=1, subdivisions=1, width=416, height=416, channels=3, momentum=0.9, decay=0.0005, angle=0, saturation = 1.5, exposure = 1.5, hue=0.1, learning\_rate=0.001, burn\_in=1000, max\_batches = 10000, policy=steps, steps=8000,9000, scales=.1,.1. 75 convolutional layers were added in a model with minimum 128 filters and maximum 512 filters.

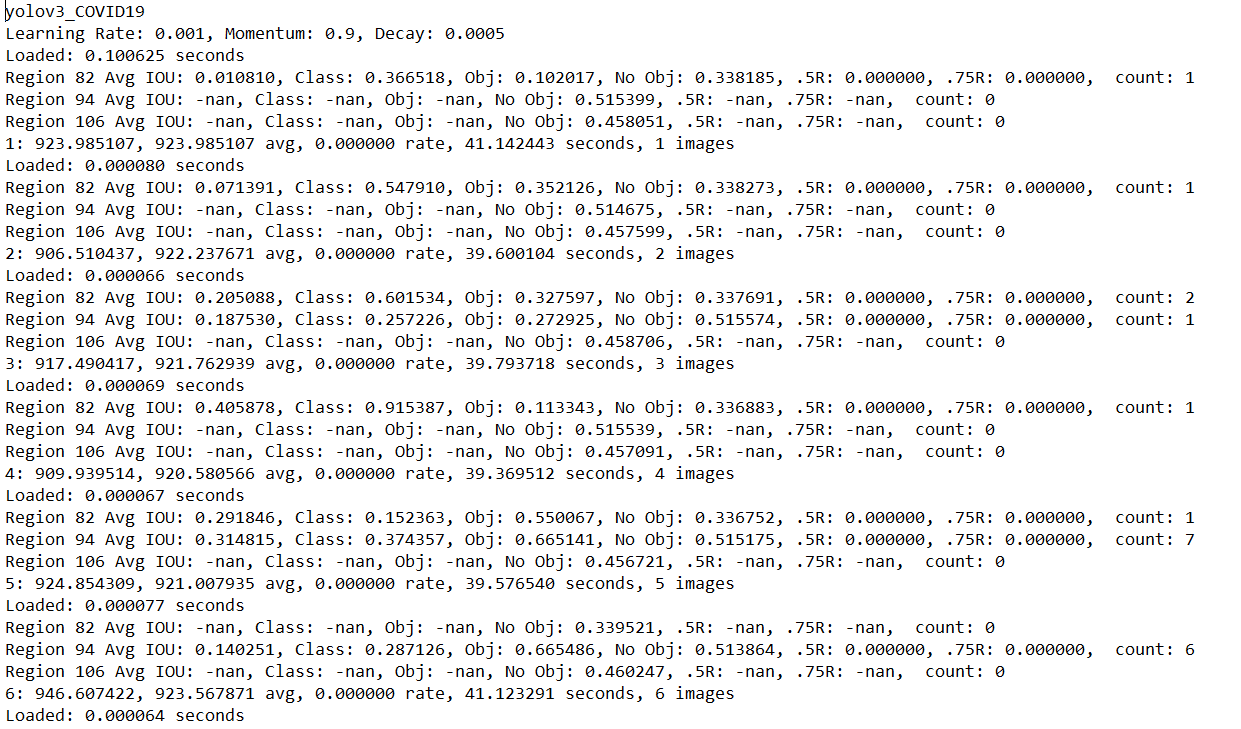
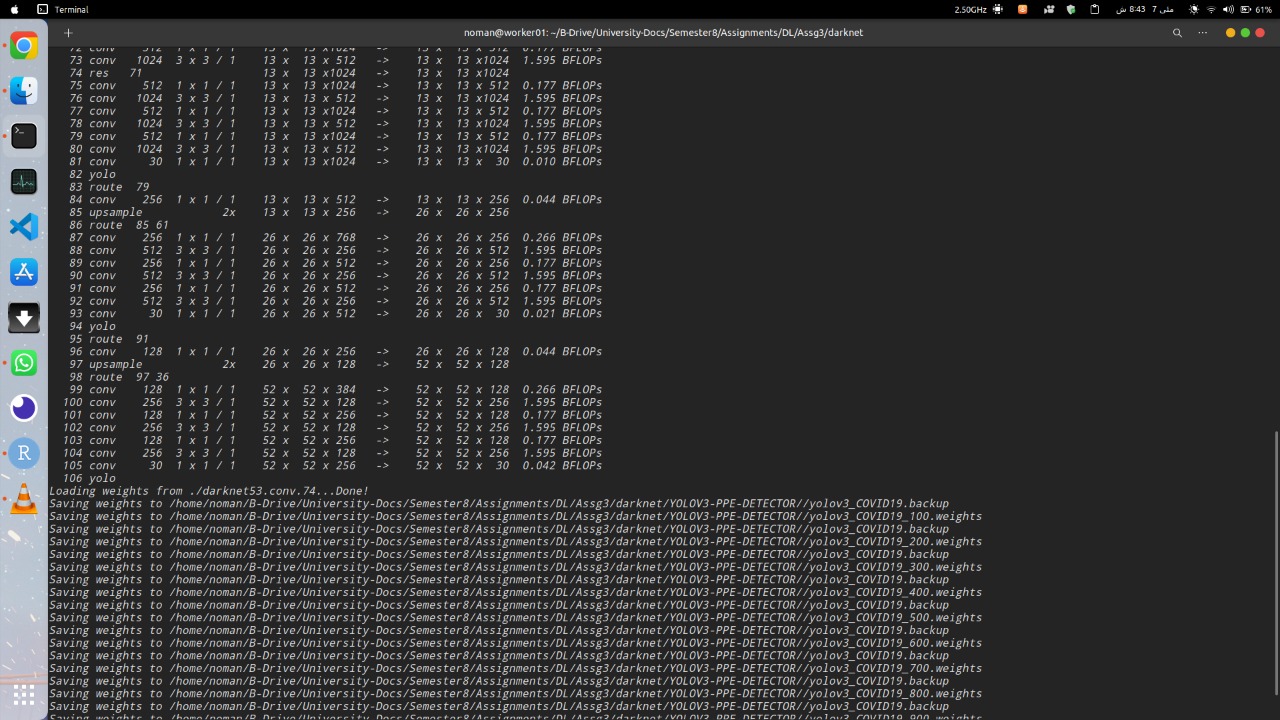
**Training:**

Darknet uses a command line interface for training and testing. First step for this is to download darknet. Command for training a yolo network with darknet is as follow:

./darknet detector train [path to obj.data] [path to .cfg file] [path to pretrained weights](optional for transfer learning) > [path to save log file]

We used partially pre trained network darknet53 conv 74 transfer learning in order save time as training this model from scratch could take alot of time. Training took around 8 hours on our Core I7 6th generation machine without gpu. And It returned 10 weights file.

****

****

**Testing:**

Testing of a trained YOLO network can be performed with the help of weights and darknet.

Command for training is as follow:

./darknet detector test [path to obj.data] [path to .cfg file] [path of trained weights] [path of test image]

We tested trained network on five random images collected from google and our network successfully detected classes within images.

Videos of test are attached with the assignment.