National University
of computer and emerging sciences

Player Identification using YOLOv3

Deep Learning
Semester Project

Submitted By:

Ayesha Zafar (19i-1983)

Muazz Amir (19k-0215)

COURSE: Deep Learning CS4045

SECTION: CS-A

SUBMITTED TO: Mr. Muhammad Muneeb Ullah

Introduction

In this project we have used YOLOv3 for training a custom dataset then the weights obtained from training are used to identify certain players in input images or videos. Flask is used to create a web application that takes the input and displays the output results.

DataSet

Dataset consists of 125 images of 4 players namely- Cristiano Ronaldo, Luka Modric, Marcelo and Sergio Ramos. There are around 30-32 images of each player. Less images affect the learning while training and too many images affect the training time. We found this number to be suitable after experimenting with lower and higher number.

Annotations are stored in txt files and are done with the help of a tool called ‘ModifiedOpenLabelling’.

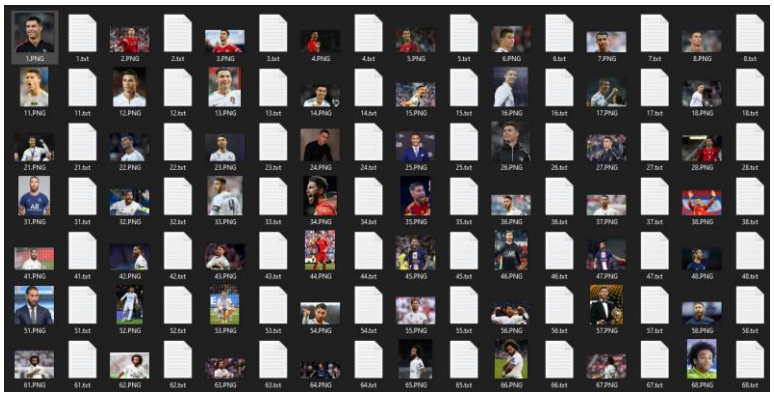

Fig i. DataSet images with annotation files

Training

Training is done with YOLOv3 and Colab GPU. Darknet’s pretrained weights were used for Transfer Learning in the process.

```
Output exceeds the size limit. Open the full output data in a text editor
Streaming output truncated to the last 5000 lines.
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 63317, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.900377), count: 1, class_loss = 0.102469, iou_loss = 0.037528, total_loss = 0.139997
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 63318, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.954099), count: 1, class_loss = 0.000008, iou_loss = 0.010730, total_loss = 0.010738
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 63319, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.869193), count: 1, class_loss = 0.000013, iou_loss = 0.045524, total_loss = 0.045537
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 63320, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.942162), count: 1, class_loss = 0.000039, iou_loss = 0.007722, total_loss = 0.007762
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 63321, rewritten_bbox = 0.000000 %
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 82 Avg (IOU: 0.909935), count: 1, class_loss = 0.000111, iou_loss = 0.064886, total_loss = 0.064998
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 94 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 106 Avg (IOU: 0.000000), count: 1, class_loss = 0.000000, iou_loss = 0.000000, total_loss = 0.000000
 total_bbox = 63322, rewritten_bbox = 0.000000 %
]2;7860/8000: loss=0.0 hours left=0.1
...
```

Fig ii. Training results with accuracy, loss and IoU

Detection

Two modes of detection are used-

- Detection on input images
- Detection on video input from web cam

In the first one, trained weights and config files are given for reference and detection is done. During the process, the person in image is compared with class values, probability of match is calculated, bounding boxes are created on found indexes and finally image with detected person is displayed.

In the second type, the video frame from web cam input are taken and same process is repeated for detection. Finally bounding boxes with probability score is displayed.

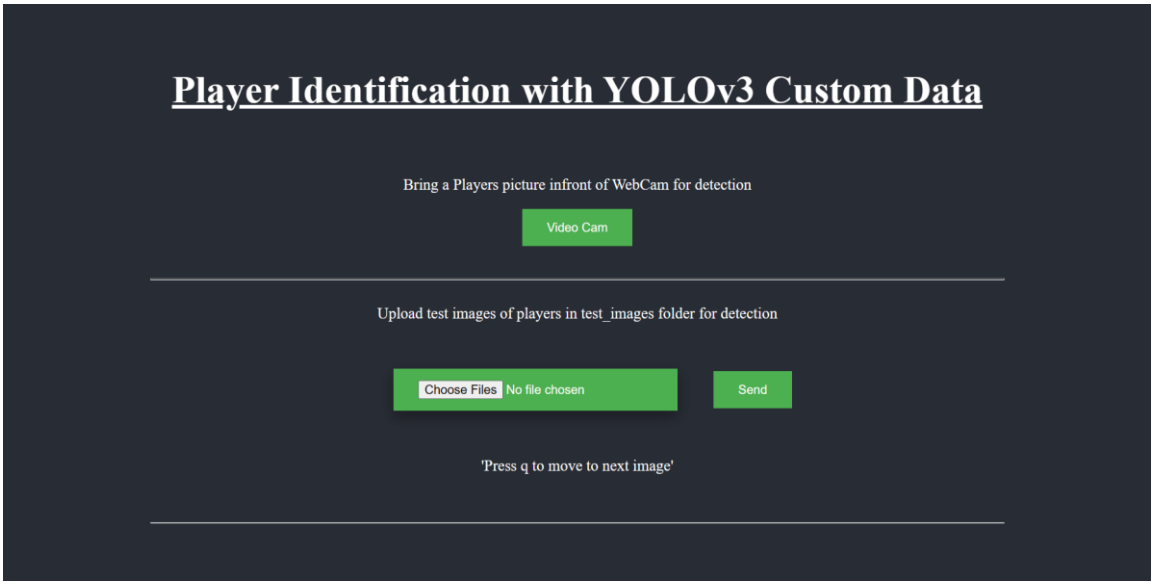

Fig iii. Web App Home Page

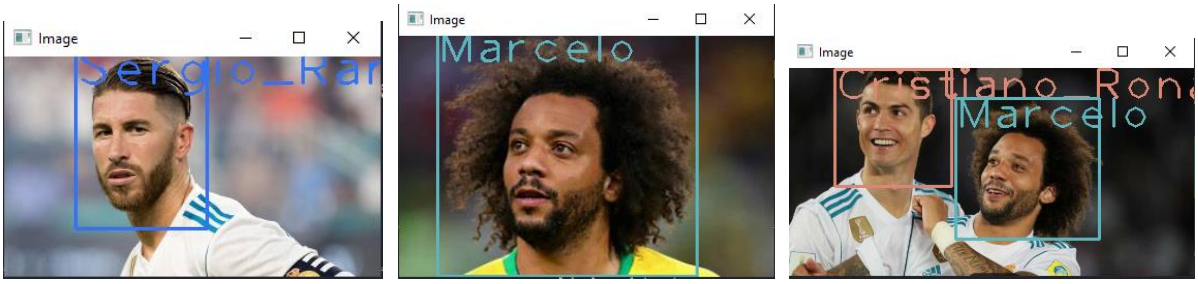

Fig iv. Detection results on test images

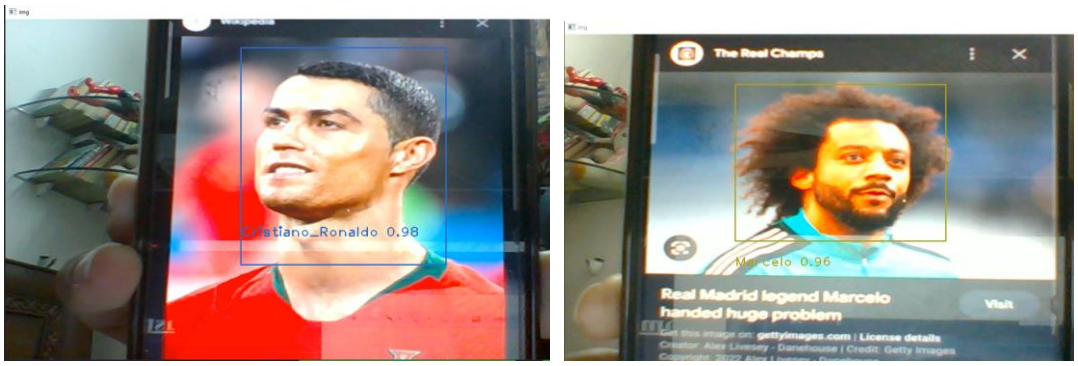

Fig v. Detection results on web cam video