Helwan University

Faculty of Computers and Artificial Intelligence

Faculty of Computers and Artificial Intelligence

Computer Science Department

2022/2023

CS 396 Selected Topics in CS-2

Research Project

Team ID No. 50

	ID	Name	Grade
1.	201900437	عبدالرحمن محمد فكري ثابت	
2.	201900435	عبدالرحمن محمد فاروق فؤاد	
3.	201900499	عمر احمد شعبان محمد	
4.	201900833	مصطفى محمود محمد عطا	
5.	201900544	عمرو مصطفى محمد أحمد	
6.	202000693	مارو عادل شمعون مقار	
7.	202000082	احمد محمود عرفات عبد الحميد	

• Paper Details

- Authors Name:

- Name: Raghad Raied Mahmood
- Name: Dr. Majid Dherar Younus
- Name: Dr. Emad Atiya Khalaf

Paper Name:

Name: Currency Detection for Visually Impaired Iraqi Banknote as a Study Case

Paper Link:

https://pdfs.semanticscholar.org/6243/5b14553080ada4234e6c5a754af0b26831e6.pdf

Publisher Name:

Turkish Journal of Computer and Mathematics Education

Year of Publishing:

2021

The implemented Algorithm: Yolov3

Results:

Table 1 shows the evaluation results of the proposed system performance. The YOLOv3 based Iraqi banknote detection and recognition system achieves 97.405% mAP on different images.

No	Denomination of the banknote	No. of test images	Average Precision AP(%)
1	250 Dinar	250	93.544%
2	500 Dinar	250	94.209%
3	1000 Dinar	250	99.995%
4	5000 Dinar	250	98.876%
5	10000 Dinar	250	99.454%
6	25000 Dinar	250	97.384%
7	50000 Dinar	250	98.374%
mAP50		97.405%	

Table 1. Evaluation Performance of The System

• Project Description

a. General Information on the selected dataset:

Dataset name: Egyptian Currency

Link: https://www.kaggle.com/datasets/egyptiris/egyptian-currency

Total Number Of Samples in the dataset:

Dataset consists of 10k Images ,
each 1000 images are loaded to a separate file

The Dimension of images:
1080*1080

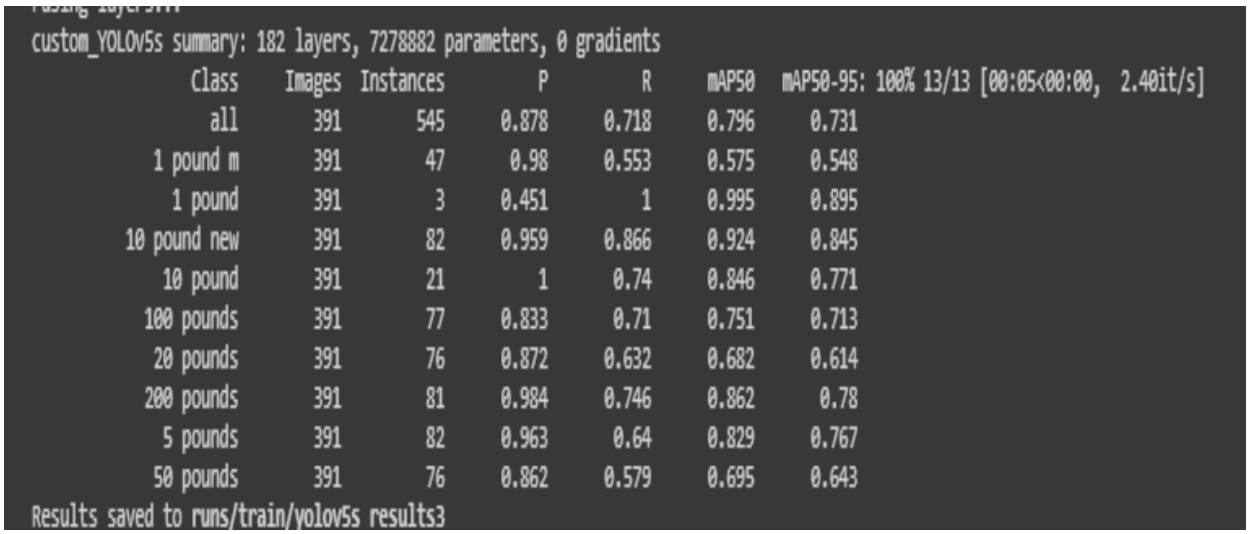
Number of Classes:

The original dataset on kaggle has 6 classes
[5, 10, 20, 50, 100, 200]

B. Implementation details

The dataset is divided into 70% train, 20 % validation, 10 % testing

And this is summary for training:



```
custom_YOLOv5s summary: 182 layers, 7278882 parameters, 0 gradients
               Class     Images  Instances          P          R      mAP50   mAP50-95: 100% 13/13 [00:05<00:00,  2.40it/s]
                 all        391        545      0.878      0.718      0.796      0.731
         1 pound m          391         47       0.98      0.553      0.575      0.548
           1 pound          391          3      0.451          1      0.995      0.895
      10 pound new          391         82      0.959      0.866      0.924      0.845
          10 pound          391         21          1       0.74      0.846      0.771
         100 pounds         391         77      0.833       0.71      0.751      0.713
          20 pounds         391         76      0.872      0.632      0.682      0.614
         200 pounds         391         81      0.984      0.746      0.862       0.78
           5 pounds         391         82      0.963       0.64      0.829      0.767
          50 pounds         391         76      0.862      0.579      0.695      0.643
Results saved to runs/train/yolov5s results3
```

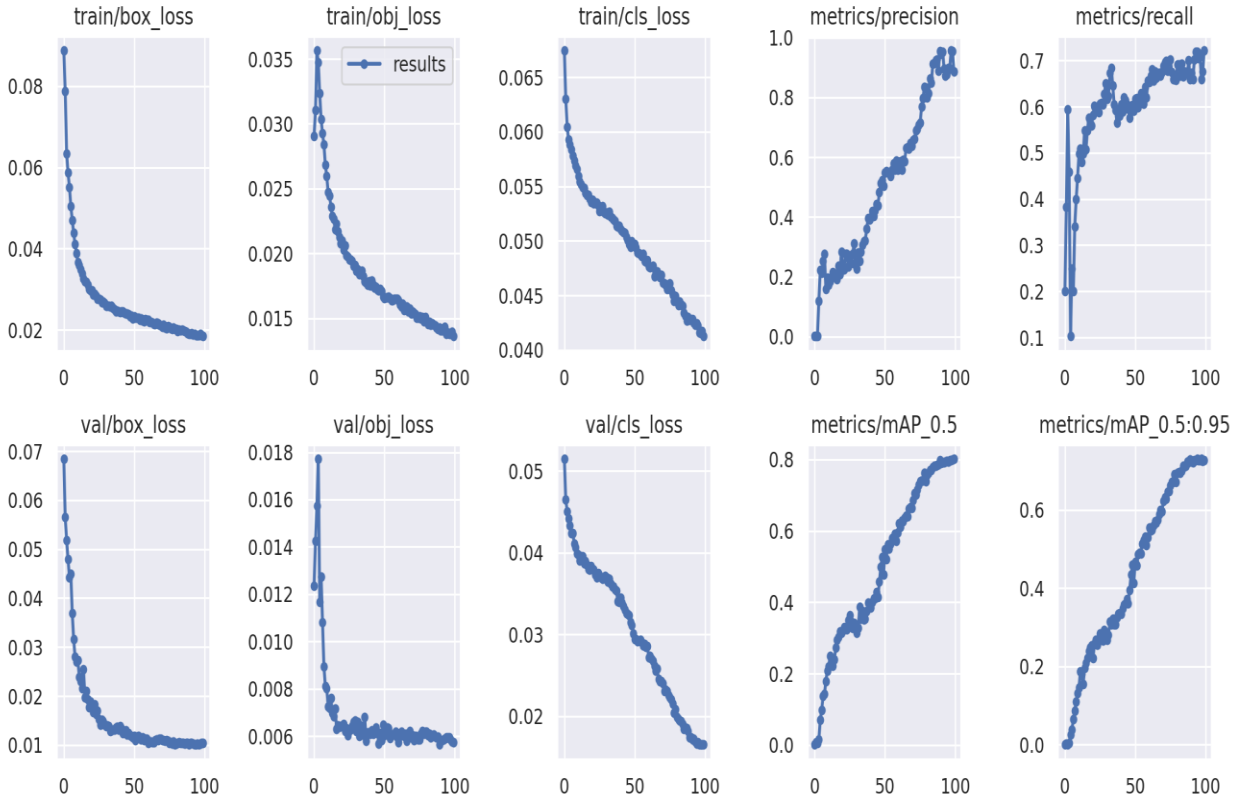
391 image for every class in the the training dataset

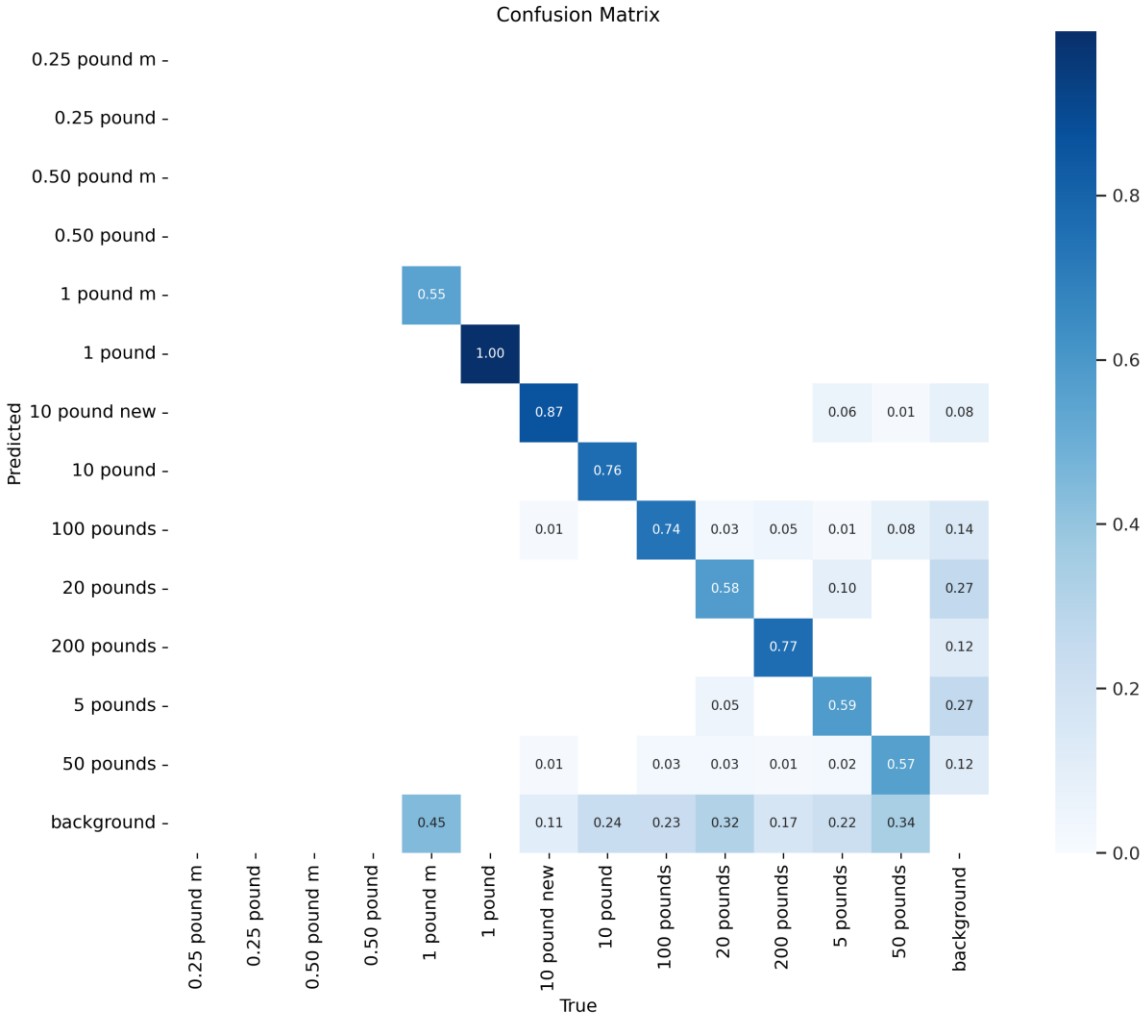
this is the model configuration used from yolov5 Rebo

```yaml
# parameters
nc: {num_classes}  # number of classes
depth_multiple: 0.33  # model depth multiple
width_multiple: 0.50  # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23]  # P3/8
  - [30,61, 62,45, 59,119]  # P4/16
  - [116,90, 156,198, 373,326]  # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]],  # 0-P1/2
   [-1, 1, Conv, [128, 3, 2]],  # 1-P2/4
   [-1, 3, BottleneckCSP, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 9, BottleneckCSP, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, BottleneckCSP, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 1, SPP, [1024, [5, 9, 13]]],
   [-1, 3, BottleneckCSP, [1024, False]],  # 9
  ]

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 6], 1, Concat, [1]],  # cat backbone P4
   [-1, 3, BottleneckCSP, [512, False]],  # 13

   [-1, 1, Conv, [256, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, 'nearest']],
   [[-1, 4], 1, Concat, [1]],  # cat backbone P3
   [-1, 3, BottleneckCSP, [256, False]],  # 17 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]],  # cat head P4
   [-1, 3, BottleneckCSP, [512, False]],  # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]],  # cat head P5
   [-1, 3, BottleneckCSP, [1024, False]],  # 23 (P5/32-large)

   [[17, 20, 23], 1, Detect, [nc, anchors]],  # Detect(P3, P4, P5)
  ]
```

Hyperparameters: SGD used with yolo algorithm

Results Details:




Confusion Matrix
Predicted
True
0.25 pound m
0.25 pound
0.50 pound m
0.50 pound
1 pound m
1 pound
10 pound new
10 pound
100 pounds
20 pounds
200 pounds
5 pounds
50 pounds
background
0.55
1.00
0.87
0.76
0.06
0.01
0.08
0.01
0.74
0.03
0.05
0.01
0.08
0.14
0.58
0.10
0.27
0.77
0.12
0.05
0.59
0.27
0.01
0.03
0.03
0.01
0.02
0.57
0.12
0.45
0.11
0.24
0.23
0.32
0.17
0.22
0.34
0.0
0.2
0.4
0.6
0.8