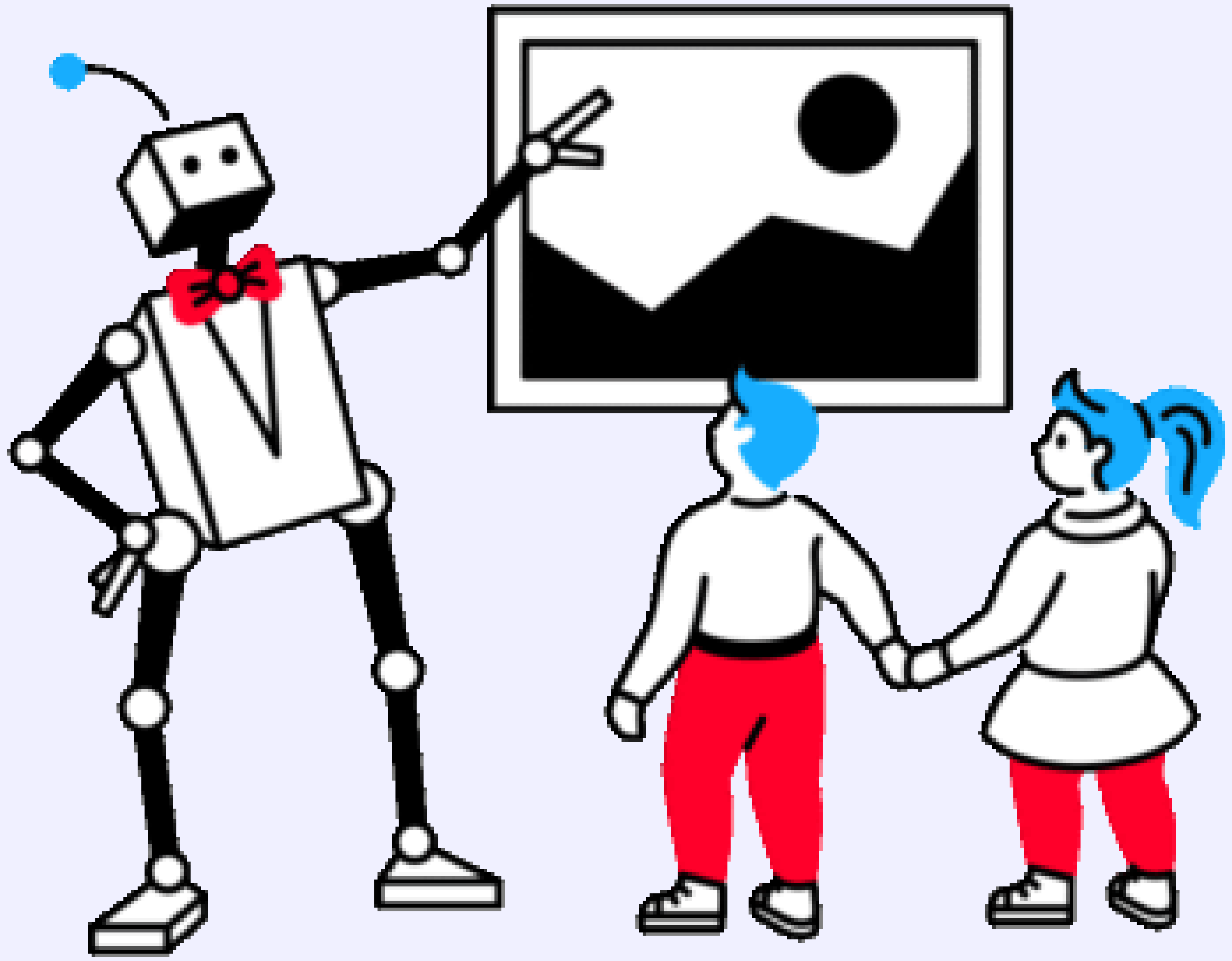
GLA UNIVERSITY
MATHURA
Recognised by UGC Under Section 2(f)
ऋते ज्ञानान्न मुक्तिः
Accredited with A Grade by NAAC

IBM
Innovation Centre for Education
B.Tech (CSE) programs
in collaboration with
GLA University, Mathura

Group Number::21

Count Number Of Objects In Image

Supervised by: Dr. Rashi Jaiswal



Ranveer Rathour

Punnya P Gaur

Prakhar Verma

Index of Presentation

01 Introduction

02 Understanding the problem Statement

03 Model Working

04 Results

05 Future Work

06 References

Introduction

- Object detection is a phenomenon in computer vision that involves the detection of various objects in digital images or videos. Some of the objects detected include people, cars, chairs, stones, buildings, and animals.

- **What is YOLO:-**YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals.

- YOLO algorithm *has gained popularity because of its superior performance over the aforementioned object detection techniques*.

Understanding the problem statement

- Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos.[1] Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video

- We propose a new supervised learning framework for visual object counting tasks, such as estimating the number of cells in a microscopic image or the number of humans in surveillance video frames. We focus on the practically-attractive case when the training images are annotated with dots (one dot per object)

Methodology & Pretrained Algorithms.

RCNN(2014):Works on region based CNN.

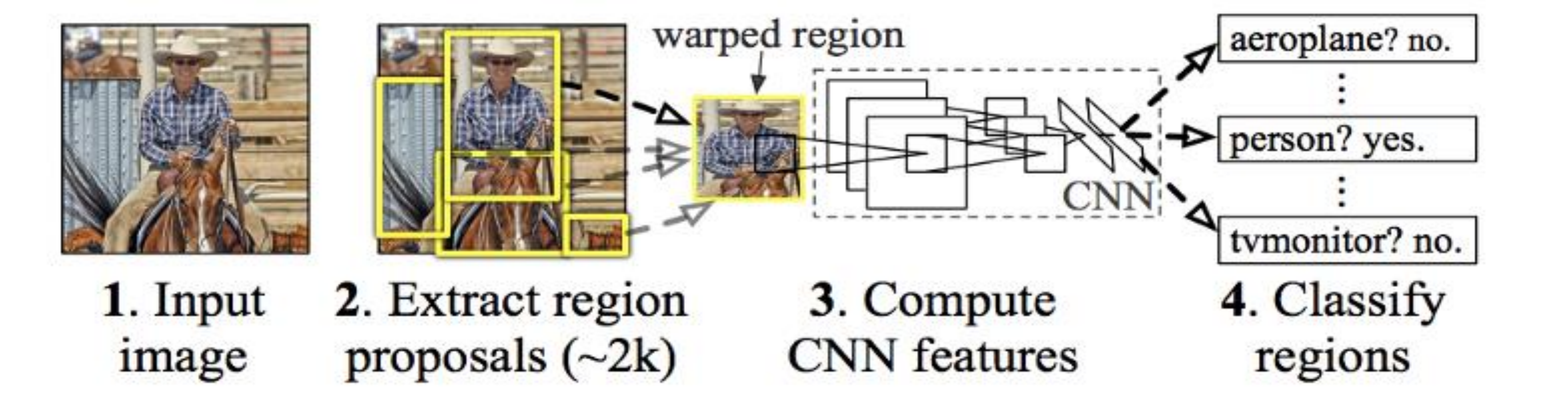
FAST R-CNN(2015): Selective search for regions of Interest.

FASTER R-CNN(2016):Region Proposal Network(RPN)

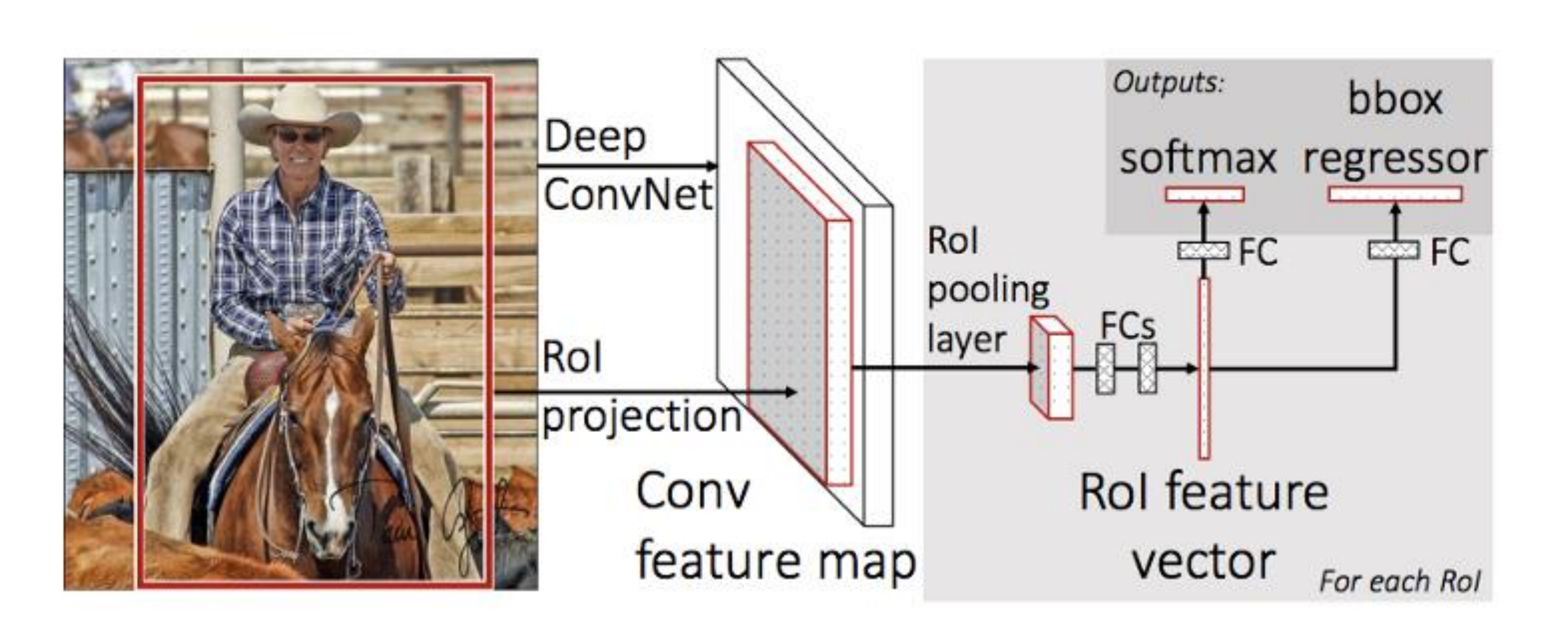
SSD(2016):Completely eliminates proposal generation and subsequent pixel and encapsulate all computation in a single network.

YOLO(2016): Accomplish Object Detection via fixed grid regression

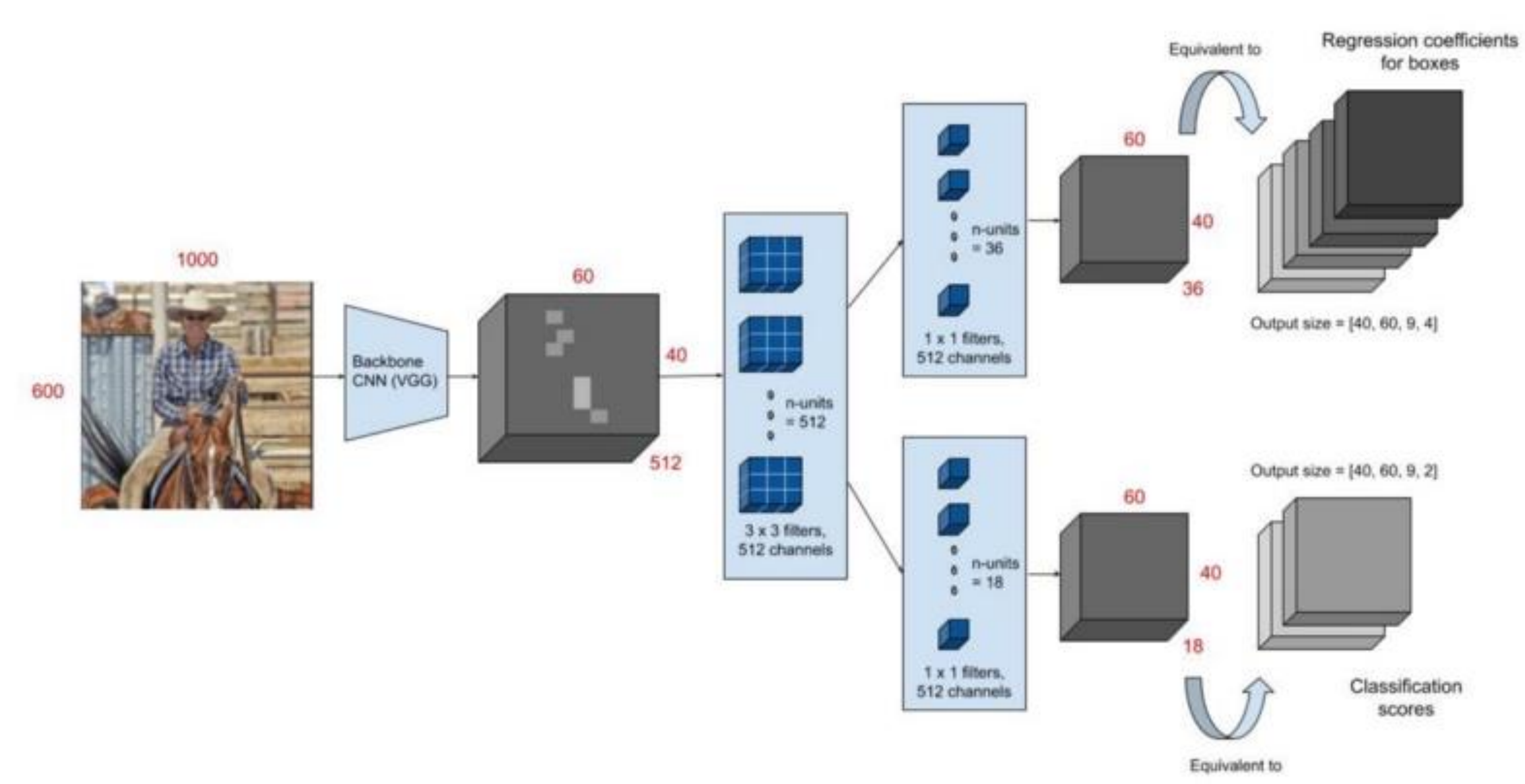
Region-Convolutional Neural Network



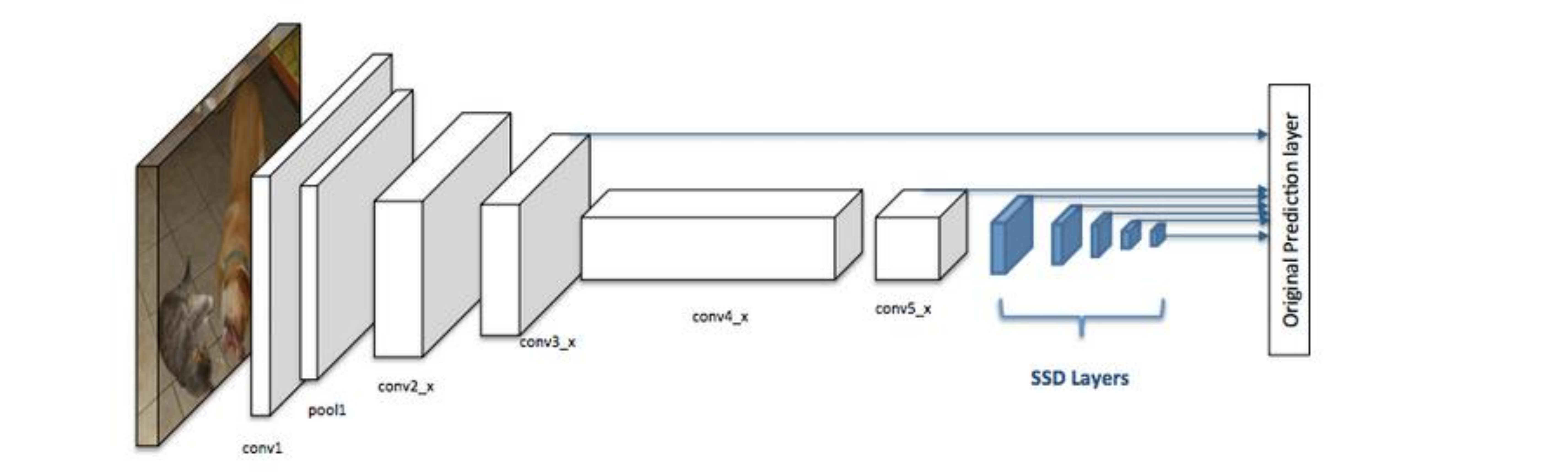
FAST R-CNN


Deep
ConvNet
RoI
projection
Conv
feature map
RoI
pooling
layer
FCs
Outputs:
softmax
bbox
regressor
FC
FC
RoI feature
vector
For each RoI

FASTER R-CNN


1000
600
Backbone CNN (VGG)
60
40
512
n-units = 512
3 x 3 filters, 512 channels
n-units = 36
1 x 1 filters, 512 channels
60
40
36
Equivalent to
Regression coefficients for boxes
Output size = [40, 60, 9, 4]
n-units = 18
1 x 1 filters, 512 channels
60
40
18
Output size = [40, 60, 9, 2]
Classification scores
Equivalent to

SSD: Single Shot



YOLO : YOU LOOK ONCE

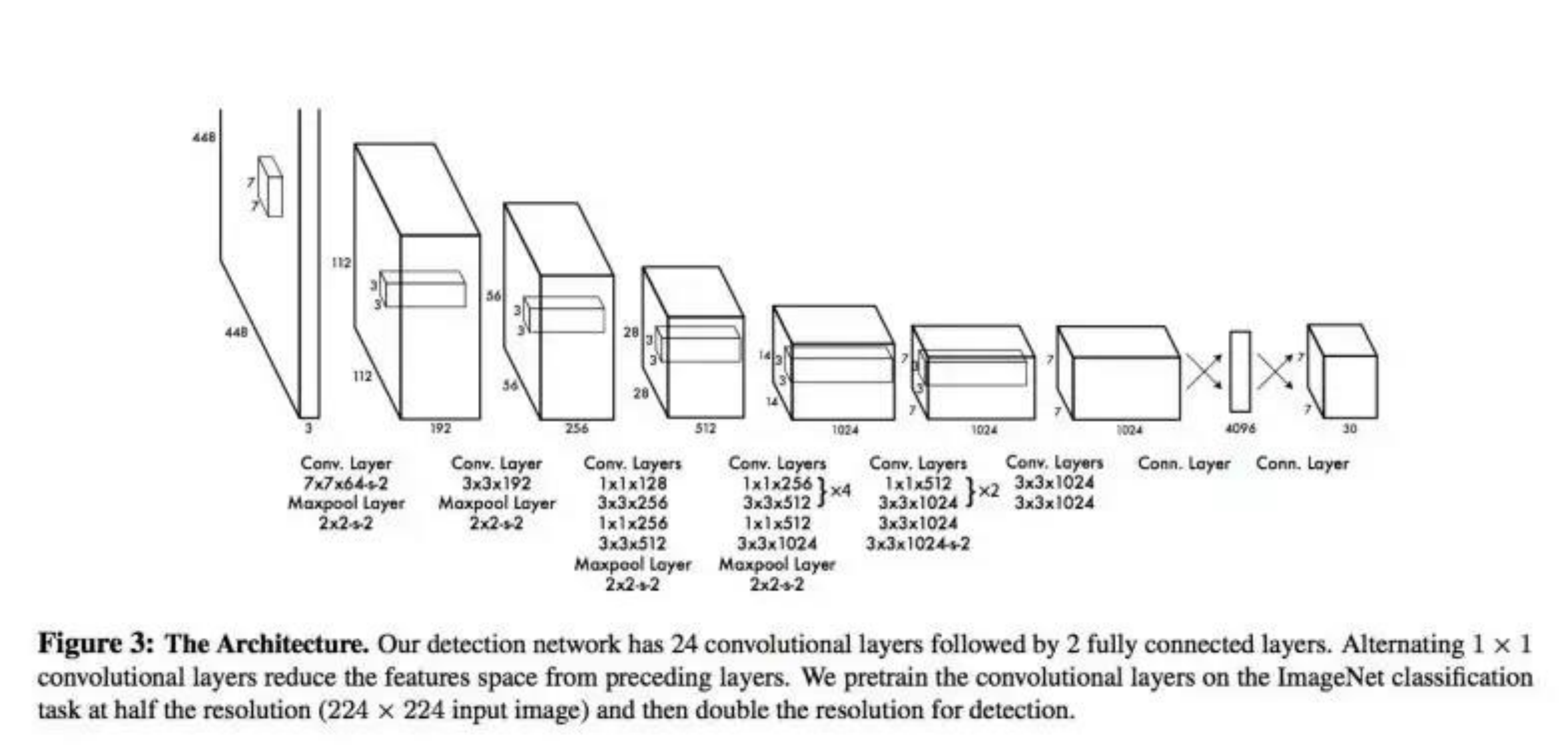
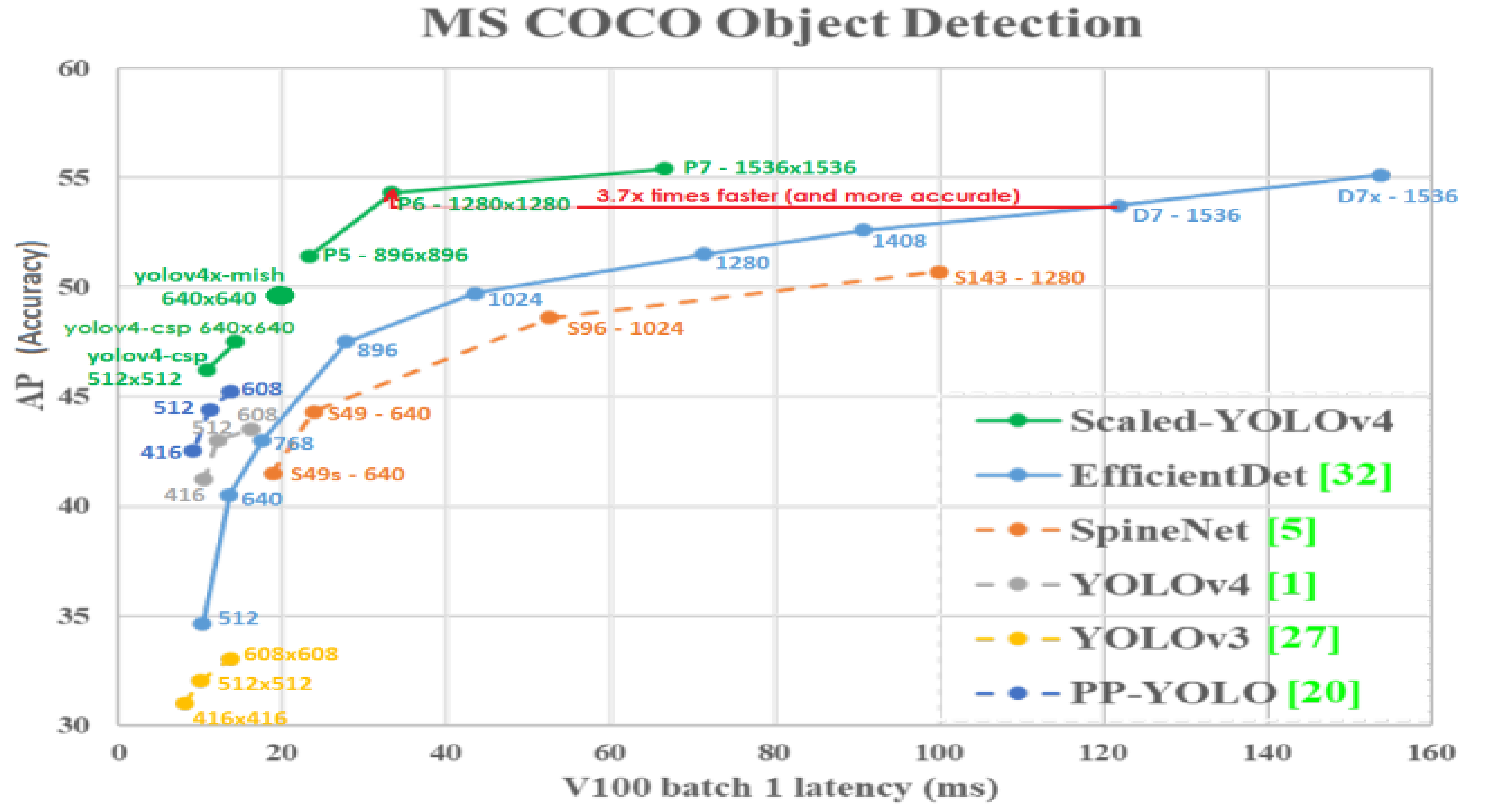
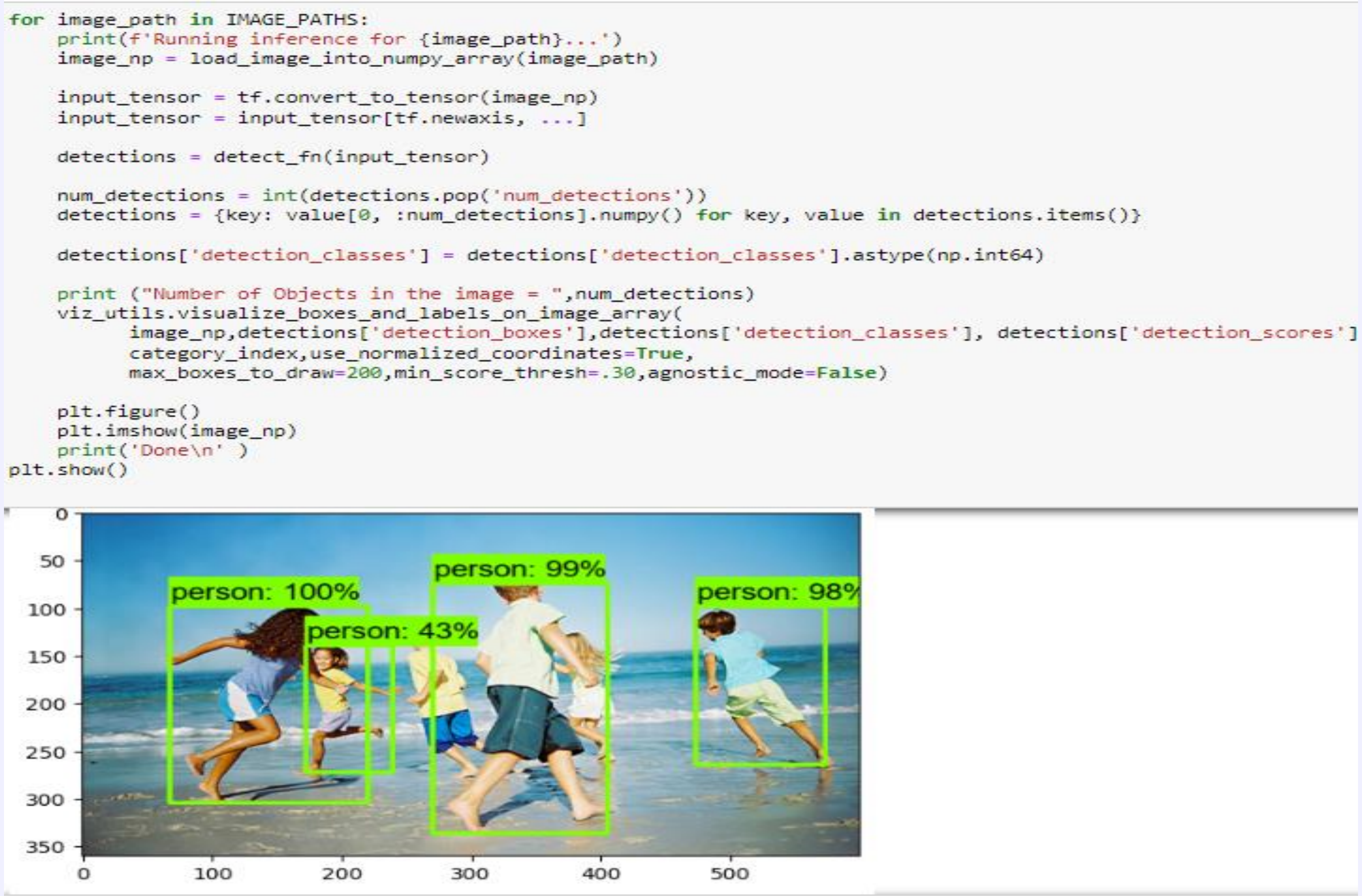


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

Difference b/w Yolo & Other Models



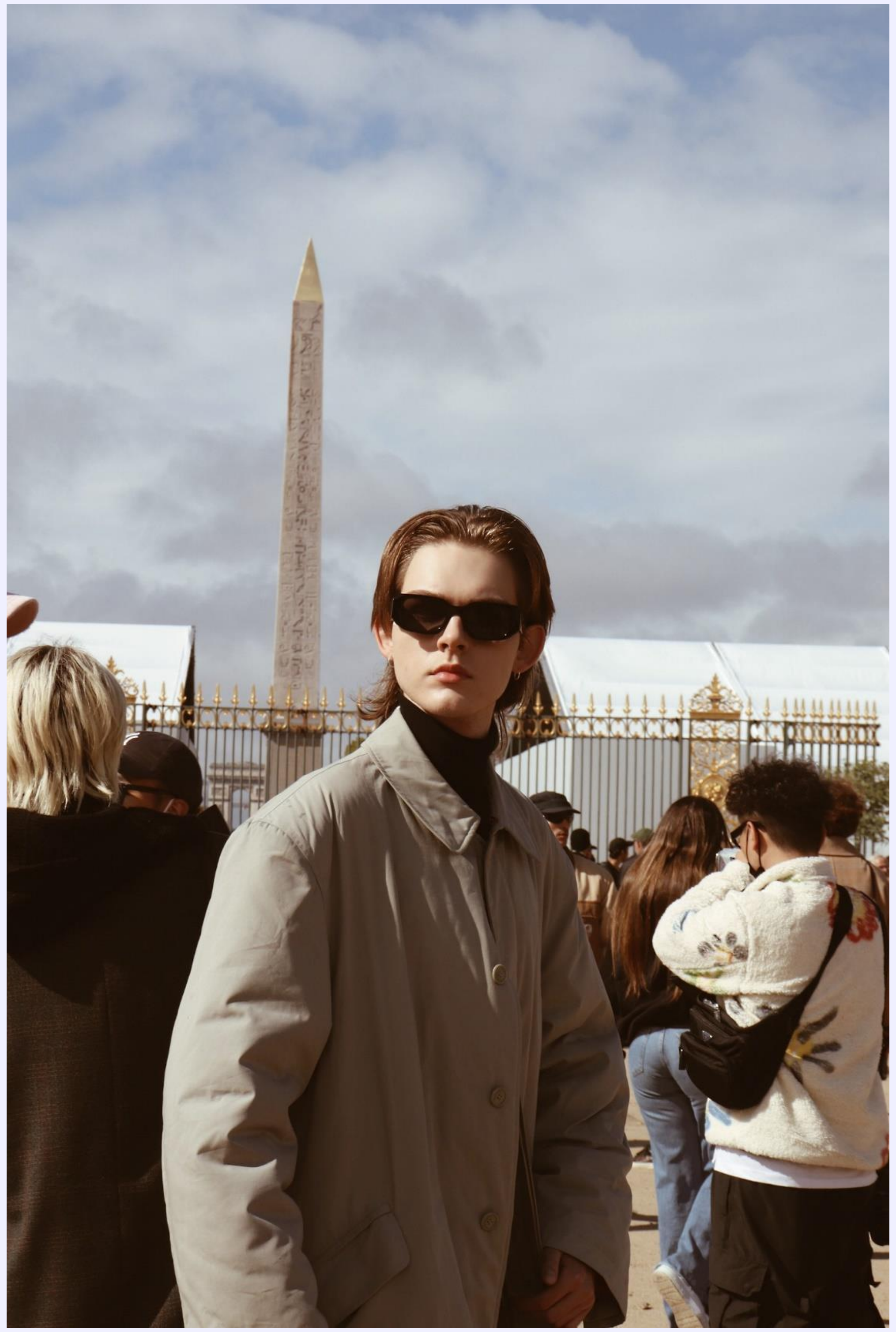
CODE

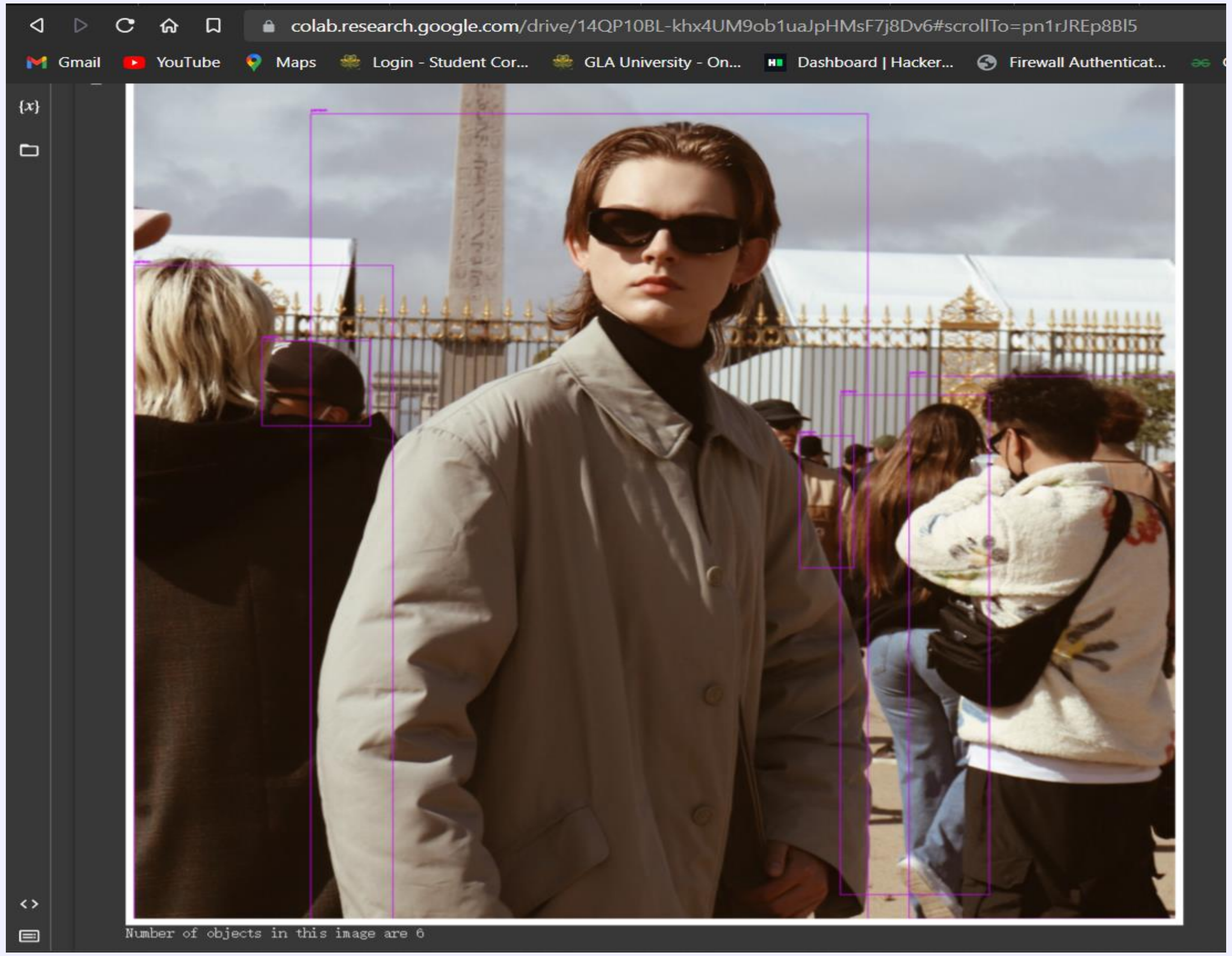


```
for image_path in IMAGE_PATHS:
    print(f'Running inference for {image_path}...')
    image_np = load_image_into_numpy_array(image_path)

    input_tensor = tf.convert_to_tensor(image_np)
    input_tensor = input_tensor[tf.newaxis, ...]

    detections = detect_fn(input_tensor)

    num_detections = int(detections.pop('num_detections'))
    detections = {key: value[0, :num_detections].numpy() for key, value in detections.items()}

    detections['detection_classes'] = detections['detection_classes'].astype(np.int64)

    print ("Number of Objects in the image = ",num_detections)
    viz_utils.visualize_boxes_and_labels_on_image_array(
          image_np,detections['detection_boxes'],detections['detection_classes'], detections['detection_scores']
          category_index,use_normalized_coordinates=True,
          max_boxes_to_draw=200,min_score_thresh=.30,agnostic_mode=False)

    plt.figure()
    plt.imshow(image_np)
    print('Done\n' )
plt.show()
```

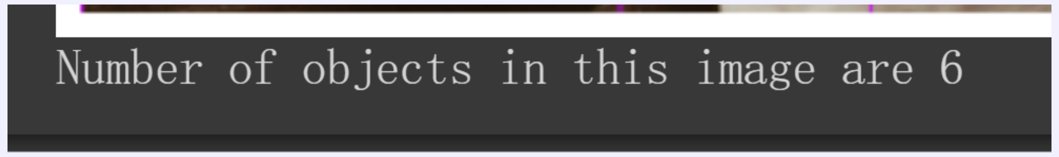
Results

Original Image



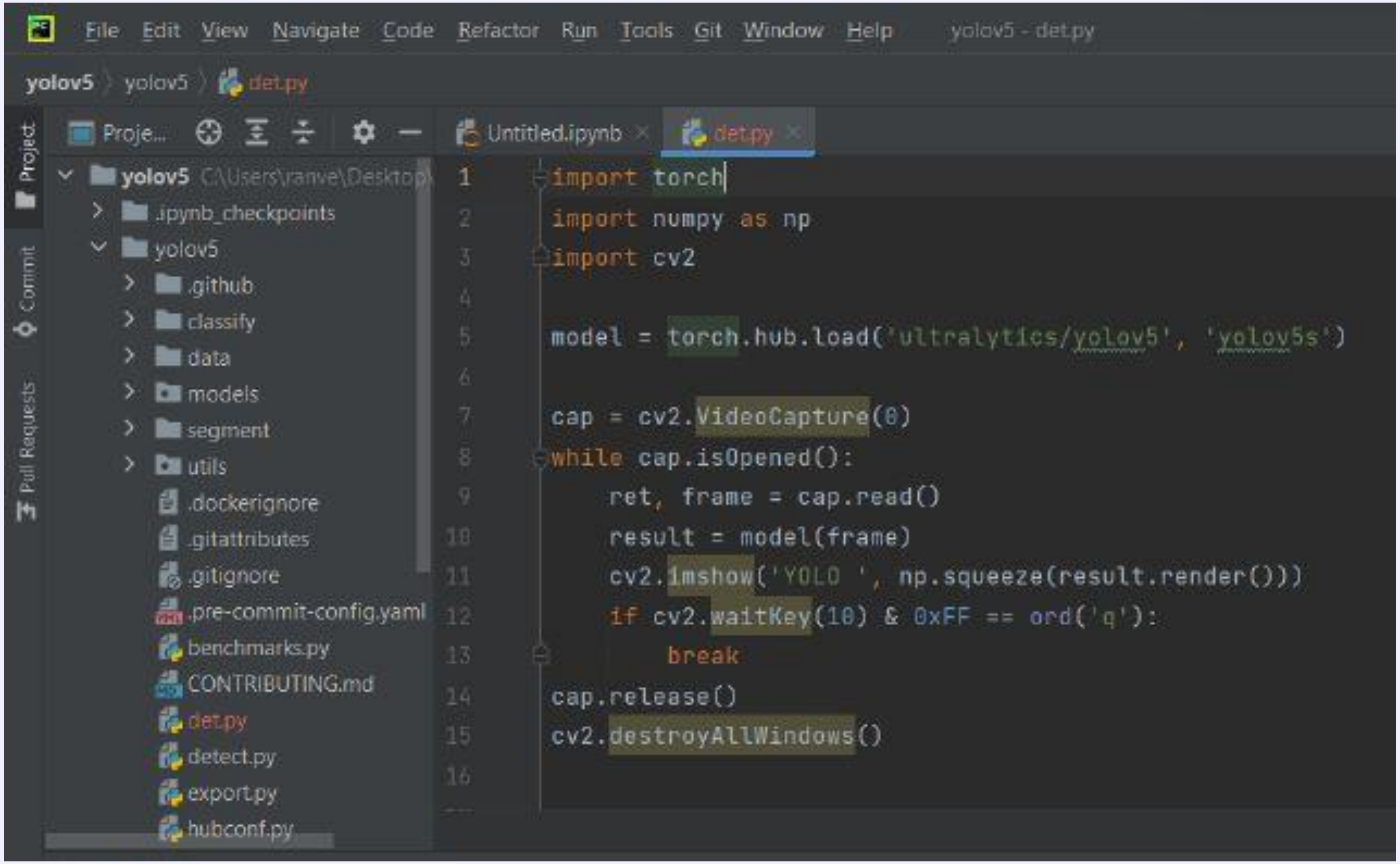


```
Number of objects in this image are 6
```



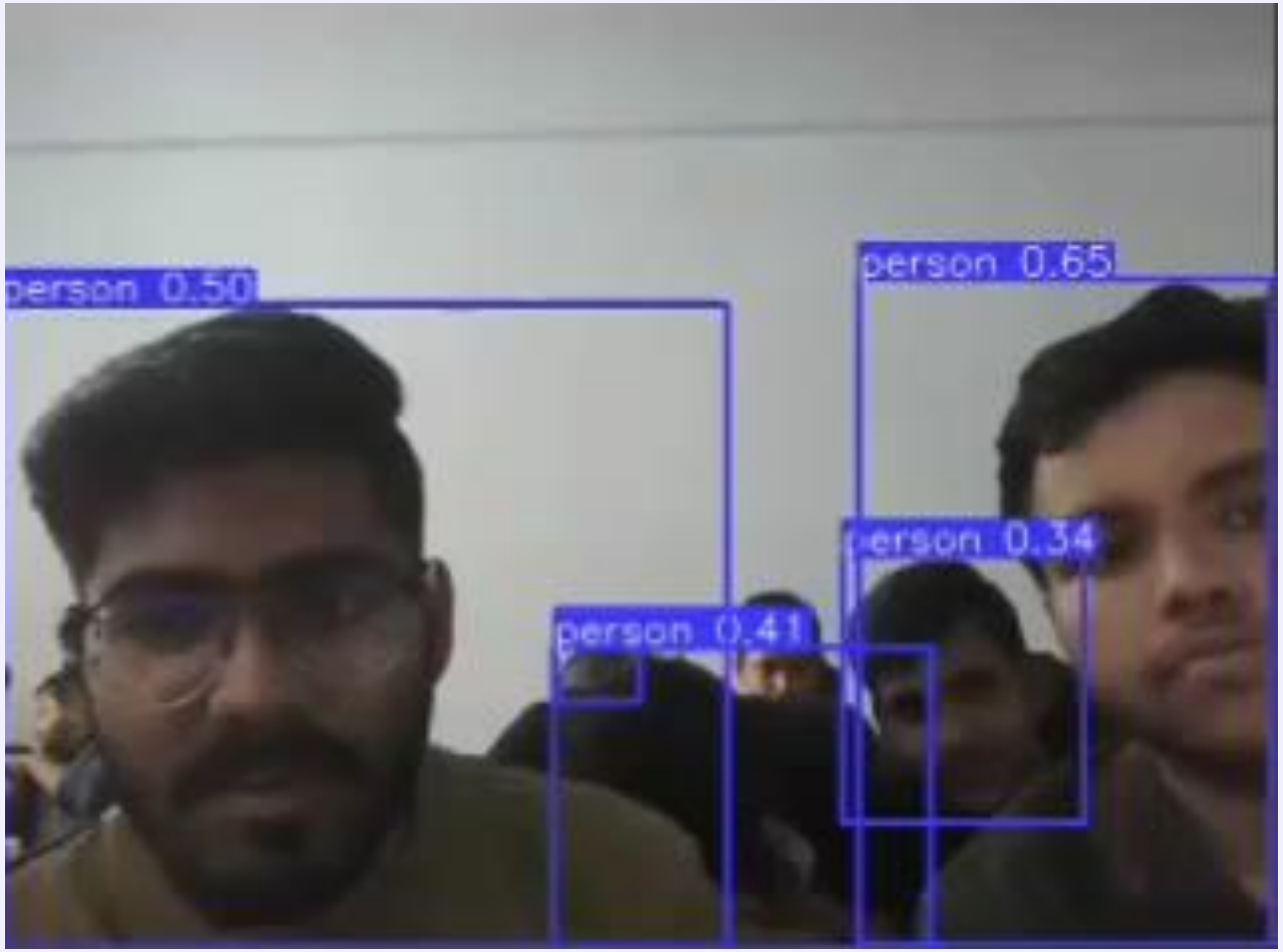
```
Number of objects in this image are 6
```

CODE

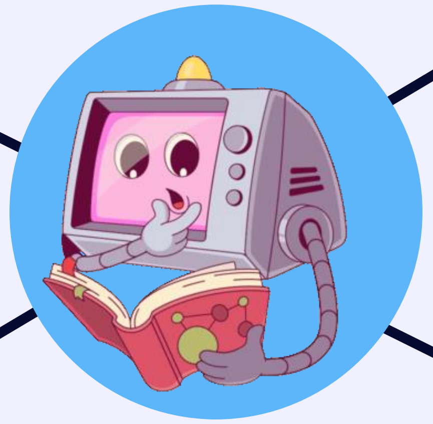


```python
import torch
import numpy as np
import cv2

model = torch.hub.load('ultralytics/yolov5', 'yolov5s')

cap = cv2.VideoCapture(0)
while cap.isOpened():
    ret, frame = cap.read()
    result = model(frame)
    cv2.imshow('YOLO ', np.squeeze(result.render()))
    if cv2.waitKey(10) & 0xFF == ord('q'):
        break
cap.release()
cv2.destroyAllWindows()
```

Results



Future Scope



Security and surveillance

Object detection is used in security and surveillance to identify and track objects in a given area. This can be used for security purposes, such as identifying potential threats, or for tracking the movements of people or objects in a pre-defined space (perimeter monitoring). Object detection is also an integral part of many facial recognition systems.

Smart city and traffic management

YOLOv7 enables object detection as used in traffic management systems to detect vehicles and pedestrians at intersections.

Visual AI in Healthcare

Hospitals and clinics use it to detect and track medical equipment, supplies, and patients. This helps to keep track of everything that is going on in the hospital (patient movement), allows for more efficient inventory management, and improves patient safety.

Autonomous Vehicles

Object detection is a key technology for self-driving cars where it is used to detect other vehicles, pedestrians, and obstacles automatically. AI vision is also used in Aviation, for autonomous drones, asset management, or even missile technologies

References

[1] https://github.com/AlexeyAB/

[2] https://www.analyticsvidhya.com/blog/2020/04/build-your-own-object-detection-model-using-tensorflow-api/

[3] https://towardsdatascience.com/computer-vision-instance-segmentation-with-mask-r-cnn-7983502fcad1

[4] https://www.youtube.com/watch?v=ATw1Dy4p1GU

[5]https://colab.research.google.com/drive/19TUG1suBe64_LcnA_ege4MEN_iVo5hXC

[6] https://icons8.com/illustrations

Thank You