OBJECT DETECTION

Name: Prathik Vittal
USN:1BM19CS117

Name: Rohit Chadichal
USN:1BM18CS084

Name: Shreyas KS
USN:1BM18CS104

Name: Mohd Irfan
USN:1BM20CS409

Guide:Vikranth B.M

Assistant Professor

Department of Computer Science & Engineering

B.M.S. College of Engineering

OUTLINE

1. Introduction
2. Problem Statement
3. Literature Survey
4. Proposed Methodology
5. Tools used for implementation
6. Implementation
7. Testing
8. Results and Discussion
9. References

Introduction

- 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.
- Well-researched domains of object detection include face detection, pedestrian detection and many other things.
- Object detection has applications in many areas of computer vision, including image retrieval and video surveillance.
- It is widely used in computer vision tasks such as image annotation, activity recognition, face detection, face recognition, video object co-segmentation.

Problem Statement

Object detection involves detecting instances of objects from a particular class in an image. The goal of object detection is to detect all instances of objects from a known class, such as people, cars or faces in an image and real time objects.

YOLO algorithm is an important algorithms which are implemented in this project or program.

Literature Survey - 1

Research Paper Name	Abstract of the research Paper	Useful ideas/insights from the research paper
A Practice for Object Detection Using YOLO Algorithm	When we look at images or videos, we can easily locate and identify the objects of our interest within moments.Object Detection has found its application in a wide variety of domains such as video surveillance, image retrieval systems, autonomous driving vehicles and many more. Various algorithms can be used for object detection but we will be focusing on the YoloV3 algorithm. YOLO stands for "You Only Look Once". The YOLO model is very accurate and allows us to detect the objects present in the frame. YOLO follows a completely different approach. Instead of selecting some regions, it applies a neural network to the entire image to predict bounding boxes and their probabilities.YOLO is a single deep convolutional neural network that splits the input image into a set of grid cells, so unlike image classification or face detection, each grid cell in YOLO algorithm will have an associated vector in the output that tells us if an object exists in that grid cell, the class of that object, the predicted bounding box for that object.	gained an overview of object detection and the YOLO algorithm Gone through the main reasons why the yolo algorithm is important Understand the concept of vectors . how the cells the split into various grids

Literature Survey - 2

Research Paper Name	Abstract of the research Paper	Useful ideas/insights from the research paper
Real-Time Object Detection with Yolo	You Only Look Once: Unified, Real-Time Object Detection, by Joseph Redmon. Their prior work is on detecting objects using a regression algorithm. To get high accuracy and good predictions they have proposed YOLO algorithm in this paper.. Understanding of Object Detection Based on CNN Family and YOLO, by Juan Du. In this paper, they generally explained about the object detection families like CNN, R-CNN and compared their efficiency and introduced YOLO algorithm to increase the efficiency. Learning to Localize Objects with Structured Output Regression, by Matthew B. Blaschko. This paper is about Object Localization. In this, they used the Bounding box method for localization of the objects to overcome the drawbacks of the sliding window method.	-To get high accuracy and good predictions they have proposed YOLO algorithm in this paper -Understanding of Object Detection Based on CNN Family and YOLO, by Juan Du -how labels as marked -how the vector matrix in initialized -why and how the anchor boxes are needed?

Literature Survey - 3

Research Paper Name	Abstract of the research Paper	Useful ideas/insights from the research paper
You Only Look Once: Unified, Real-Time Object Detection	We present YOLO, a new approach to object detection. Prior work on object detection repurposes classifiers to perform detection. Instead, we frame object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. Our unified architecture is extremely fast. Our base YOLO model processes images in real-time at 45 frames per second. A smaller version of the network, Fast YOLO, processes an astounding 155 frames per second while still achieving double the mAP of other real-time detectors. Compared to state-of-the-art detection systems, YOLO makes more localization errors but is less likely to predict false positives on background. Finally, YOLO learns very general representations of objects. It outperforms other detection methods, including DPM and R-CNN, when generalizing from natural images to other domains like artwork.	-unifying the separate components of object detection into a single neural network -IOU truth prediction how coordinates of bounded can be determined

Literature Survey - 4

Research Paper Name	Abstract of the research Paper	Useful ideas/insights from the research paper
A Review of Yolo Algorithm Developments	Object detection techniques are the foundation for the artificial intelligence field. This research paper gives a brief overview of the You Only Look Once (YOLO) algorithm and its subsequent advanced versions. Through the analysis, we reach many remarks and insightful results. The results show the differences and similarities among the YOLO versions and between YOLO and Convolutional Neural Networks (CNNs). The central insight is the YOLO algorithm improvement is still ongoing.This article briefly describes the development process of the YOLO algorithm, summarizes the methods of target recognition and feature selection, and provides literature support for the targeted picture news and feature extraction in the financial and other fields. Besides, this paper contributes a lot to YOLO and other object detection literature.	-differences between the versions of yolo algorithm -which is suitable version -methods of target recognition

Literature Survey - 5

Research Paper Name	Abstract of the research Paper	Useful ideas/insights from the research paper
Visual Object Detection and Tracking using YOLO and SORT	Over the past two decades, computer vision has received a great deal of coverage. Visual object tracking is one of the most important areas of computer vision. Tracking objects is the process of tracking over time a moving object (or several objects). The purpose of visual object tracking in consecutive video frames is to detect or connect target objects. In this paper, we present analysis of tracking-by-detection approach which include detection by YOLO and tracking by SORT algorithm. This paper has information about custom image dataset being trained for 6 specific classes using YOLO and this model is being used in videos for tracking by SORT algorithm. Recognizing a vehicle or pedestrian in an ongoing video is helpful for traffic analysis. The goal of this paper is for analysis and knowledge of the domain.	-visual object tracking is done on videos by training detector for custom dataset consisting of 800 images for specific 6 classes -the breaking video and images into frames -Identifying static and dynamic objects

Image Classification

Is this a dog or a person?

Neural Network Output

Dog = 1
Person = 0

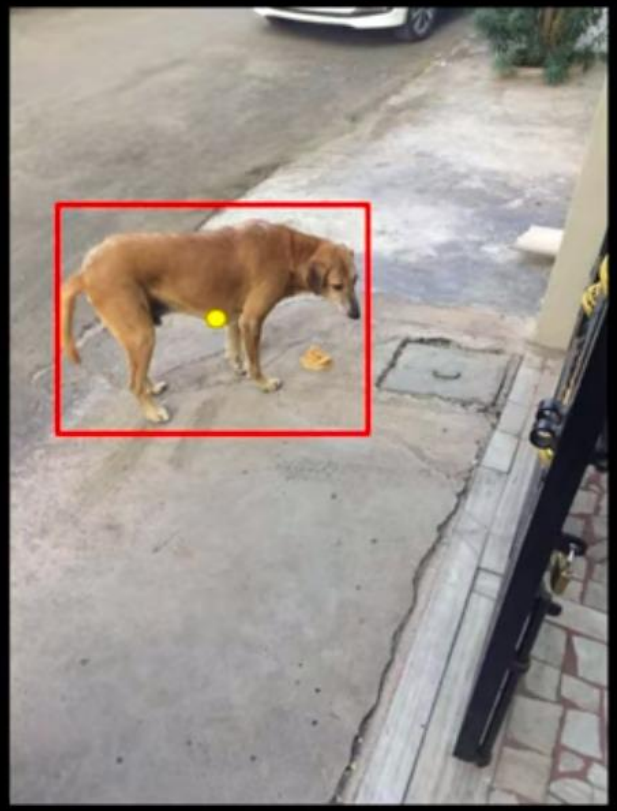
Object Localization

Where exactly is the dog in this image?

Neural Network Output

Dog = 1
Person = 0
+
Bounding Box

Object Localization

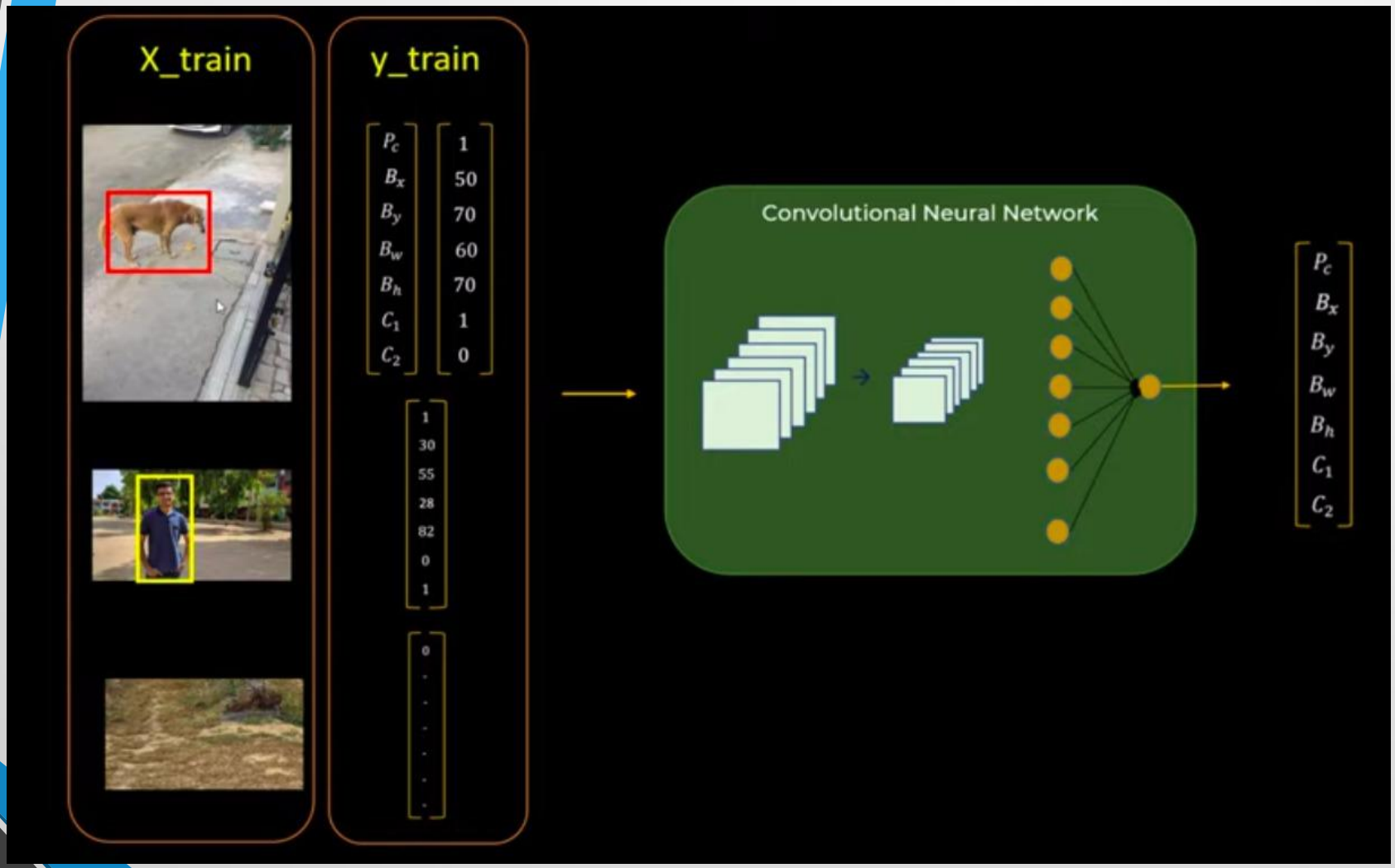


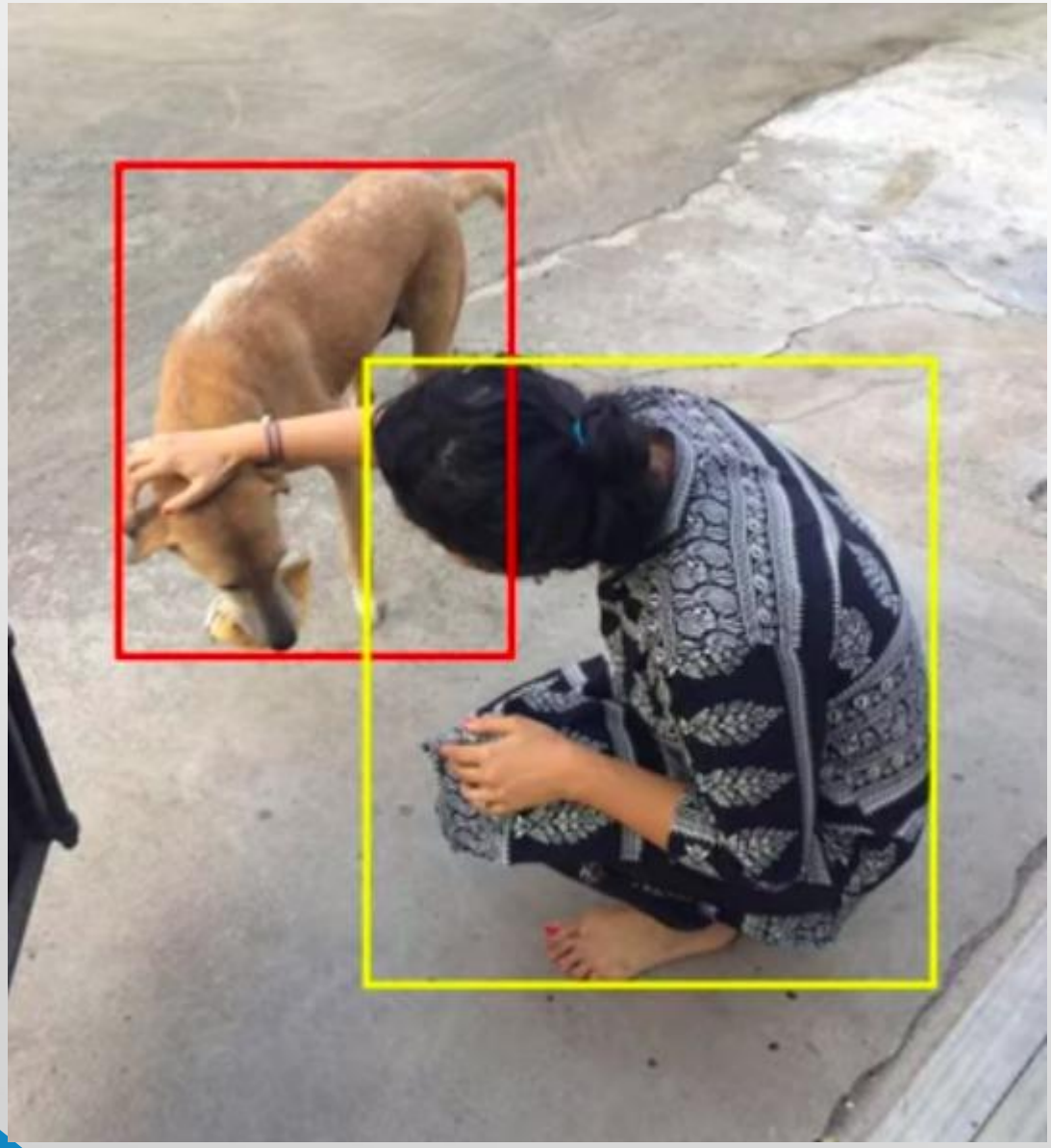
$$\begin{bmatrix} P_c \\ B_x \\ B_y \\ B_w \\ B_h \\ C_1 \\ C_2 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 50 \\ 70 \\ 60 \\ 70 \\ 1 \\ 0 \end{bmatrix}$$

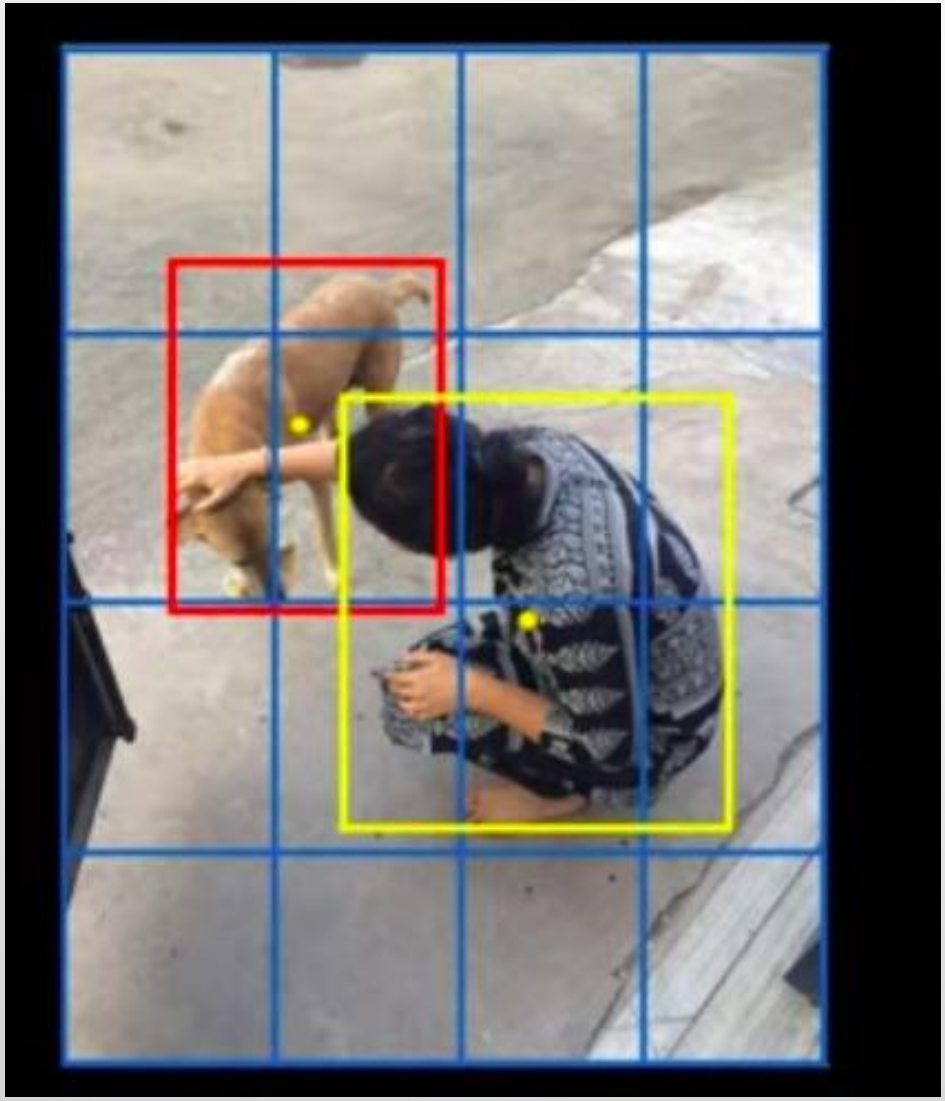
C_1 = *Dog class*
C2 = Person Class

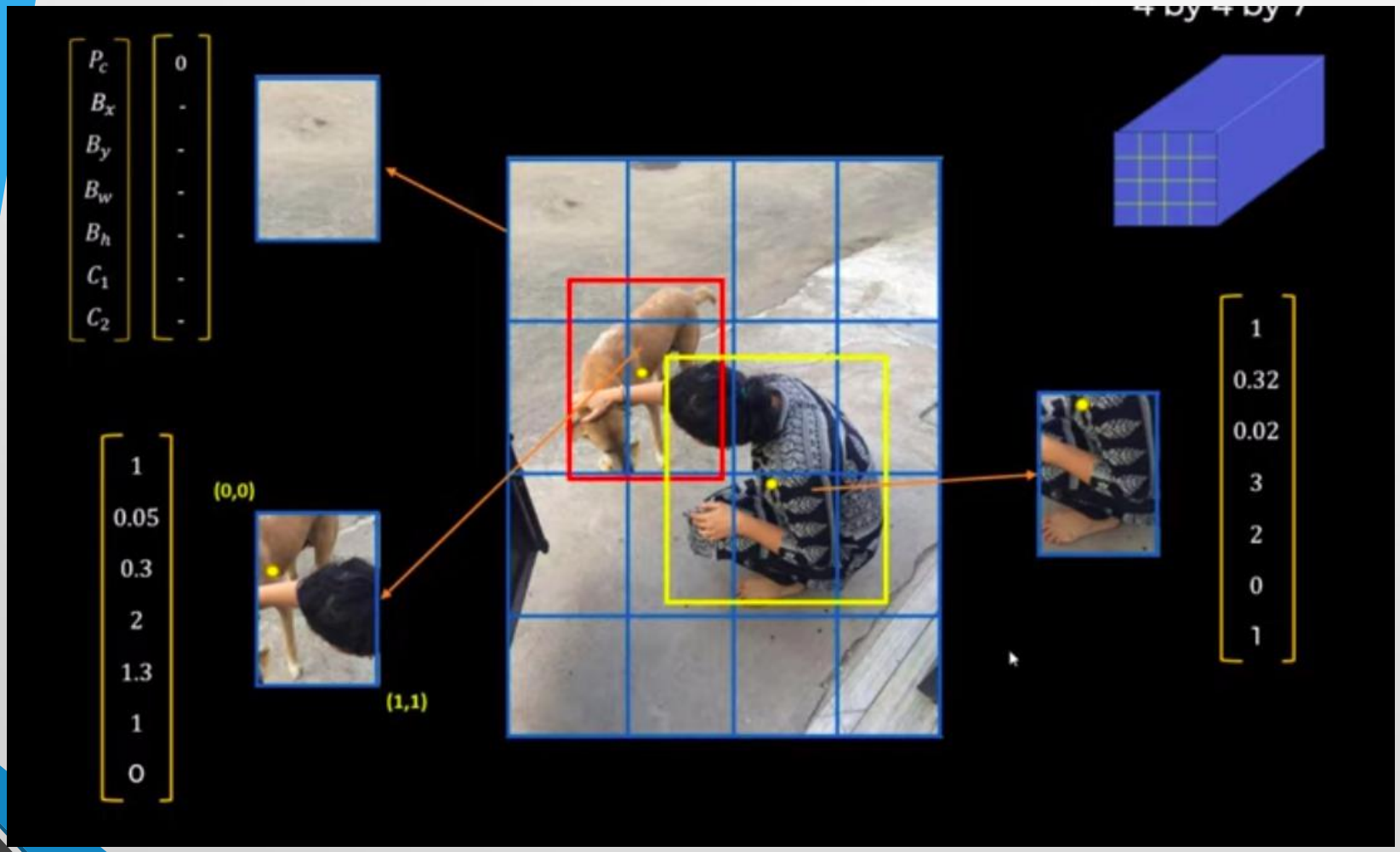
$$\begin{bmatrix} 1 \\ 30 \\ 28 \\ 28 \\ 82 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 \\ ? \\ - \\ - \\ - \\ - \\ - \end{bmatrix}$$

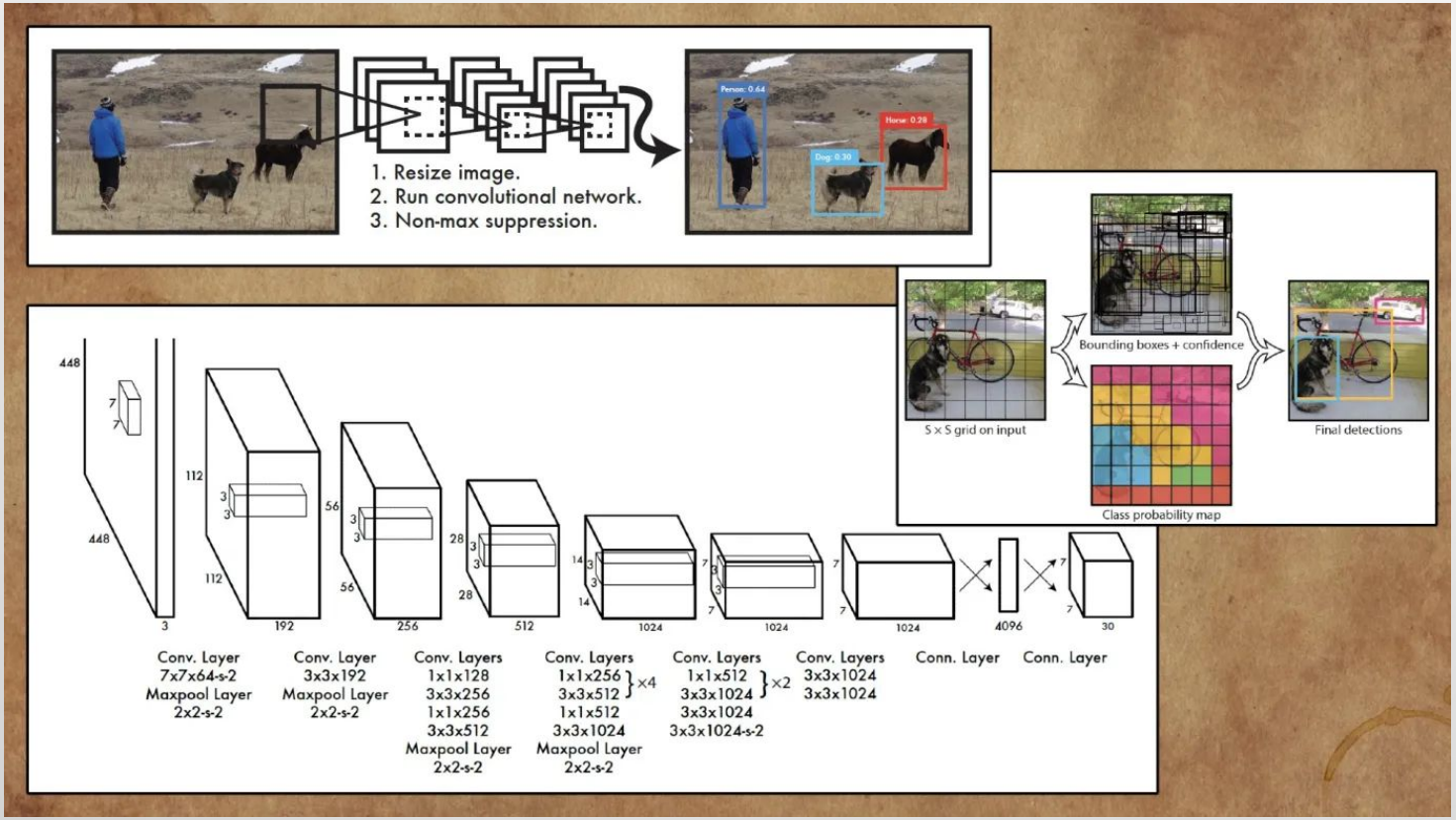

X_train
y_train
P_c 1
B_x 50
B_y 70
B_w 60
B_h 70
C_1 1
C_2 0
1 30 55 28 82 0 1
0 . . . . . .
Convolutional Neural Network
P_c B_x B_y B_w B_h C_1 C_2






4 by 4 by 7
P_c
B_x
B_y
B_w
B_h
C_1
C_2
0
-
-
-
-
-
-
1
0.05
0.3
2
1.3
1
0
(0,0)
(1,1)
1
0.32
0.02
3
2
0
1

Methodologies



Methodologies

- YOLO predicts multiple bounding boxes per grid cell.
- At training time we only want one bounding box predictor to be responsible for each object.
- We assign one predictor to be “responsible” for predicting an object based on which prediction has the highest current IOU (Intersection Over Union) with the ground truth.
- This leads to specialization between the bounding box predictors.
- Each predictor gets better at predicting certain sizes, aspect ratios, or classes of object, improving overall recall.

Methodologies

What are the criteria you are using to evaluate your method?

- Accuracy: We are using the criteria of accuracy to determine evaluation method is up to the mark

What specific hypotheses does your experiment test?

- Our experiment basically tests for different kinds of objects present in the frame .

Opencv is a vast library in python which contains n number of hypothesis and modules.

Tools Used for the Implementation

- TERMINAL : Terminals are usually faster compared to IDE's, which helps to streamline the production process. We use the terminal to run the applications
- OPENCV: This platform was built to provide a common infrastructure for computer vision applications and accelerate the development of products.
- JUPYTERLAB: We will use this to develop and debug the code. Also helps in sharing the documents.
- PACKAGES:
    1. IMUTILS: Series of functions to make basic image processing functions.
    2. NUMPY: Works in domain of linear algebra, fourier transform, and matrices and provide an array object that is up to 50x faster than traditional Python lists.
    3. ARGPARSE: User-friendly command-line interfaces, automatically generates help and usage messages and issues errors when users give the program invalid arguments.

Implementation

Real Time Object Detection:

Real-time object detection is the task of doing object detection in real-time with fast inference while maintaining a base level of accuracy.

Deep Learning Object Detection:

Object detection using deep learning provides a fast and accurate means to predict the location of an object in an image. Deep learning is a powerful machine learning technique in which the object detector automatically learns image features required for detection tasks.

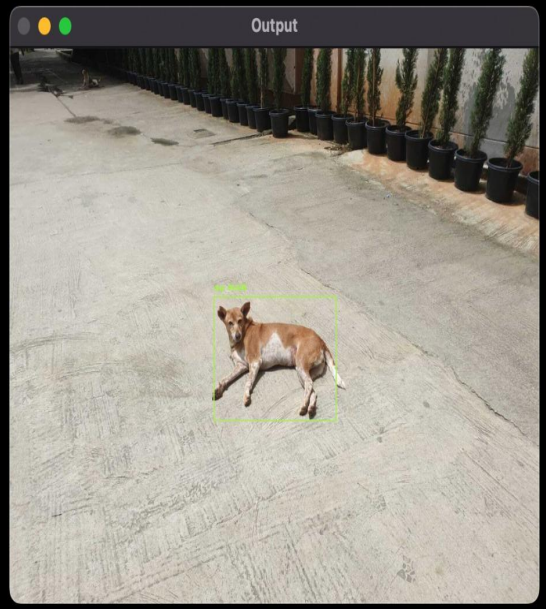
Testing

- Since testing involves the process of checking the behavior of a system with the expected behavior.
- There needs to be a metric, the metric we are considering here is human intuition. Since, humans are well versed with identifying objects and classifying them.
- We use the human capability to identify objects to test if the trained model is outputting the desired output.
- If the model outputs a erroneous output, we retrain the model.
- This process of checking and retraining is carried out until the desired outputs are achieved.

Results and Discussion

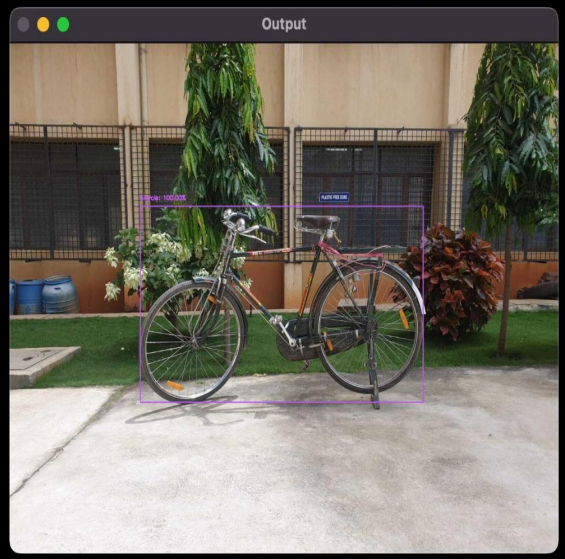
```
>  python deep_learning_object_detection.py --image i
mages/Potted_plant.jpg  --prototxt MobileNetSSD_deplo
y.prototxt.txt --model MobileNetSSD_deploy.caffemodel
[INFO] loading model...
[INFO] computing object detections...
[INFO] pottedplant: 99.98%
```


```
> python deep_learning_object_detection.py --image images/Dog2.jp
g  --prototxt MobileNetSSD_deploy.prototxt.txt --model MobileNetS
SD_deploy.caffemodel
[INFO] loading model...
[INFO] computing object detections...
[INFO] dog: 96.52%
```

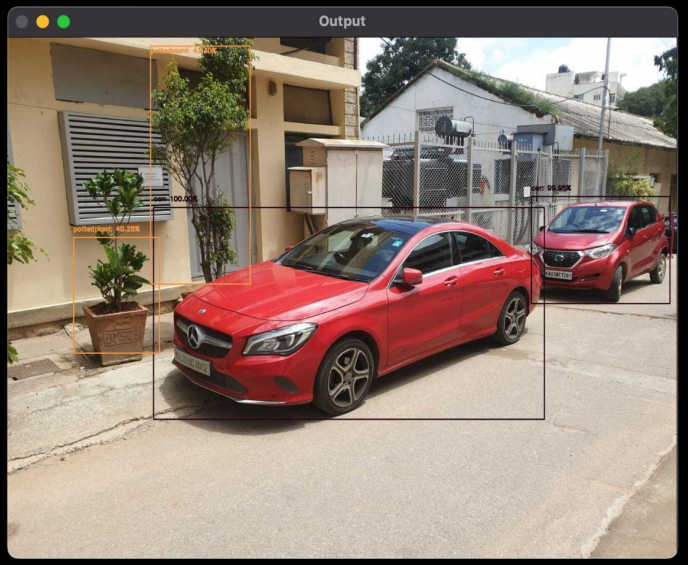


Results and Discussion

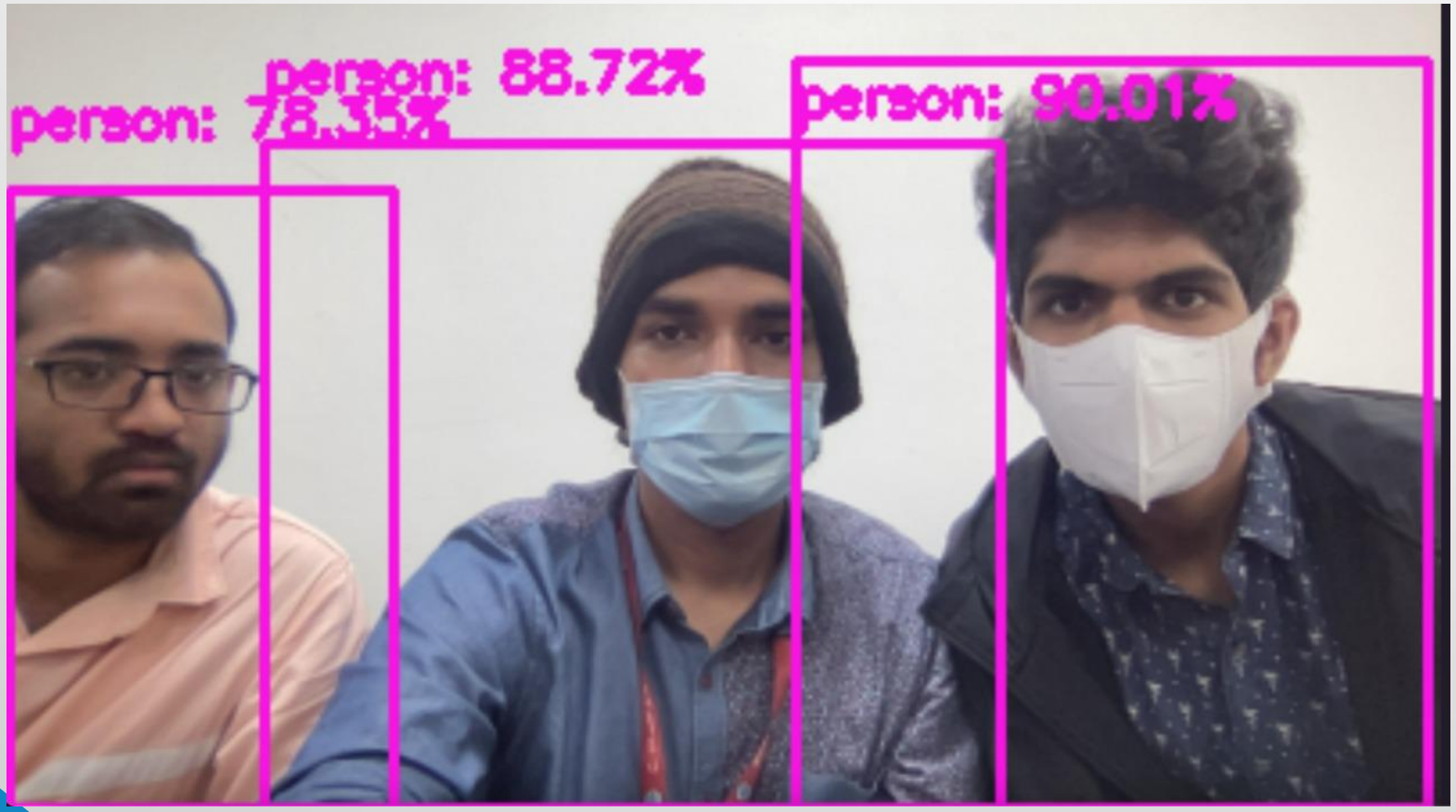
```
> python deep_learning_object_detection.py --image images/bicycle
.jpg  --prototxt MobileNetSSD_deploy.prototxt.txt --model MobileN
etSSD_deploy.caffemodel
[INFO] loading model...
[INFO] computing object detections...
[INFO] bicycle: 100.00%
```




```
> python deep_learning_object_detection.py --image images/Car.jpg
  --prototxt MobileNetSSD_deploy.prototxt.txt --model MobileNetSS
D_deploy.caffemodel
[INFO] loading model...
[INFO] computing object detections...
[INFO] car: 100.00%
[INFO] car: 99.95%
[INFO] pottedplant: 43.30%
[INFO] pottedplant: 40.28%
```



Results and Discussion



Results and Discussion

Object detection is a key ability for most computer and robot vision system. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for autofocus in smartphones) or have been integrated in assistant driving technologies, we are still far from achieving human-level performance, in particular in terms of open-world learning.

Finally, we need to consider that we will need object detection systems for nano-robots or for robots that will explore areas that have not been seen by humans, such as depth parts of the sea or other planets, and the detection systems will have to learn to new object classes as they are encountered. In such cases, a real-time open-world learning ability will be critical.

References

- https://www.researchgate.net/publication/351311739_A_Practice_for_Object_Detection_Using_YOLO_Algorithm (2021)
- https://www.ijeat.org/wp-content/uploads/papers/v8i3S/C11240283S19.pdf (2019)
- https://arxiv.org/pdf/1506.02640.pdf (2016)
- https://www.sciencedirect.com/science/article/pii/S1877050922001363 (2020 and 2021)
- https://d1wqtxts1xzle7.cloudfront.net/61397049/visual-object-detection-and-tracking-using-yolo-and-sort-IJERTV8IS11034320191202-89770-hldlwl-with-cover-page-v2.pdf?Expires=1658146355&Signature=ekD0mAHbiSi6s19IVadmvxyOXZMGKH4Z08P2KrFVHQkMOj6QF8b6CSxIyQxSVVkBtpXFleb8yAVGWK7dSCgLA69E2-QD-Le-yQXUCRSqJ6mLNzvz-Ft~un9phkoj6bv7mKCT1KiTOSKwHjdXaCc2t4sGwu8l7eIJPO49HcDwNkcwsEUaL1LlNhXQhPaZh9vC9yWv85mTLf5hBheWnd4iXFtg9W~xa891qPKp-egrvyweI7gnL7zLTM4-N0CKS2zLgCPft5lhAJYYgrRwpkirbeEuyYz4KFF0Rd3v4F7skUw2FUhxtx1tZAiQKNNsQ51coCv4TuC5KI4mcju0Y5kVOA__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA (2019)

THANK YOU