OBJECT DETECTION

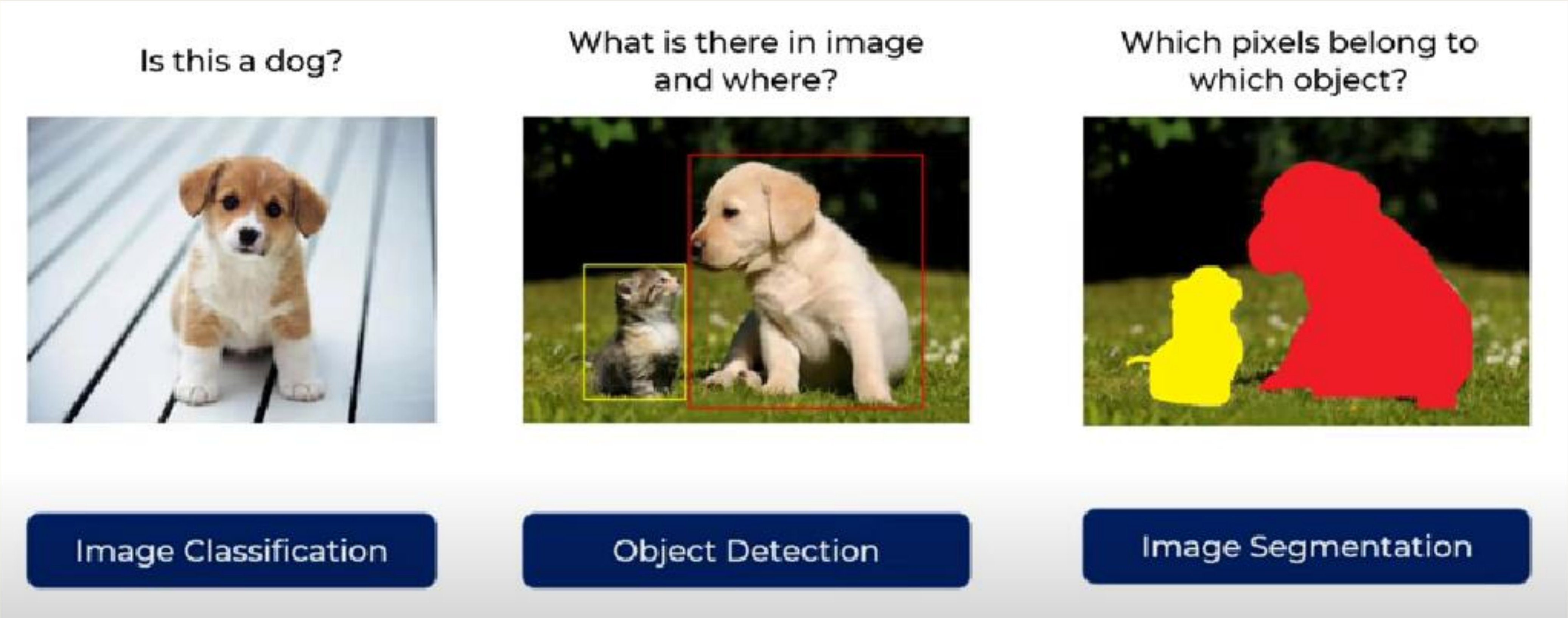
ICT4201: DIP

Introduction

- Object detection is a crucial task in computer vision that involves identifying and locating objects within an image or video.
- This task is fundamental for various applications, including autonomous driving, video surveillance, and medical imaging.

What is Object Detection?

- Object detection is a computer vision technique that combines image classification and object localization to identify and locate objects within an image. Unlike image classification, which assigns a single label to an entire image, object detection identifies multiple objects and their locations using bounding boxes.
- Key Concepts in Object Detection
  - *Object Localization:* *This involves determining the location of objects within an image by drawing bounding boxes around them.*
  - *Object Classification:* *This involves identifying the category to which the detected object belongs.*
  - *Bounding Boxes:* *These are rectangular boxes used to define the location of objects within an image*


Is this a dog?
What is there in image and where?
Which pixels belong to which object?
Image Classification
Object Detection
Image Segmentation

Applications of Object Detection

- Object detection has a wide range of applications, including:
  - *Autonomous Vehicles: Detecting pedestrians, vehicles, and other obstacles to navigate safely.*
  - *Video Surveillance: Identifying suspicious activities or objects in real-time to enhance security.*
  - *Medical Imaging: Detecting anomalies or diseases in medical scans to assist in diagnosis.*
  - *Retail: Monitoring inventory and customer behavior in stores*

Challenges in Object Detection

- **Imbalanced Datasets**: In many domains, negative samples (images without the object of interest) vastly outnumber positive samples, making it difficult to train accurate models.
- **Domain Adaptation**: Models trained on one type of data may not perform well on another due to differences in data distribution. Techniques like unsupervised domain adaptation are used to address this issue.
- **Real-Time Processing**: Achieving real-time performance while maintaining high accuracy is a significant challenge, especially in applications like autonomous driving and video surveillance.

Traditional Image Processing Techniques

- Traditional Image Processing Techniques
- Traditional image processing techniques for object detection often involve feature extraction followed by classification. Some of the notable methods include:
  - Histogram of Oriented Gradients (HOG): This technique extracts gradient orientation histograms from an image and uses them as features for object detection. It is particularly effective for human detection.
  - Viola-Jones Algorithm: Widely used for face detection, this algorithm uses Haar-like features and a cascade of boosted classifiers to detect objects in real-time.
  - Bag of Features Model: Similar to the bag of words model in text processing, this approach represents an image as an unordered collection of features, which are then used for classification.

Understanding HOG Features

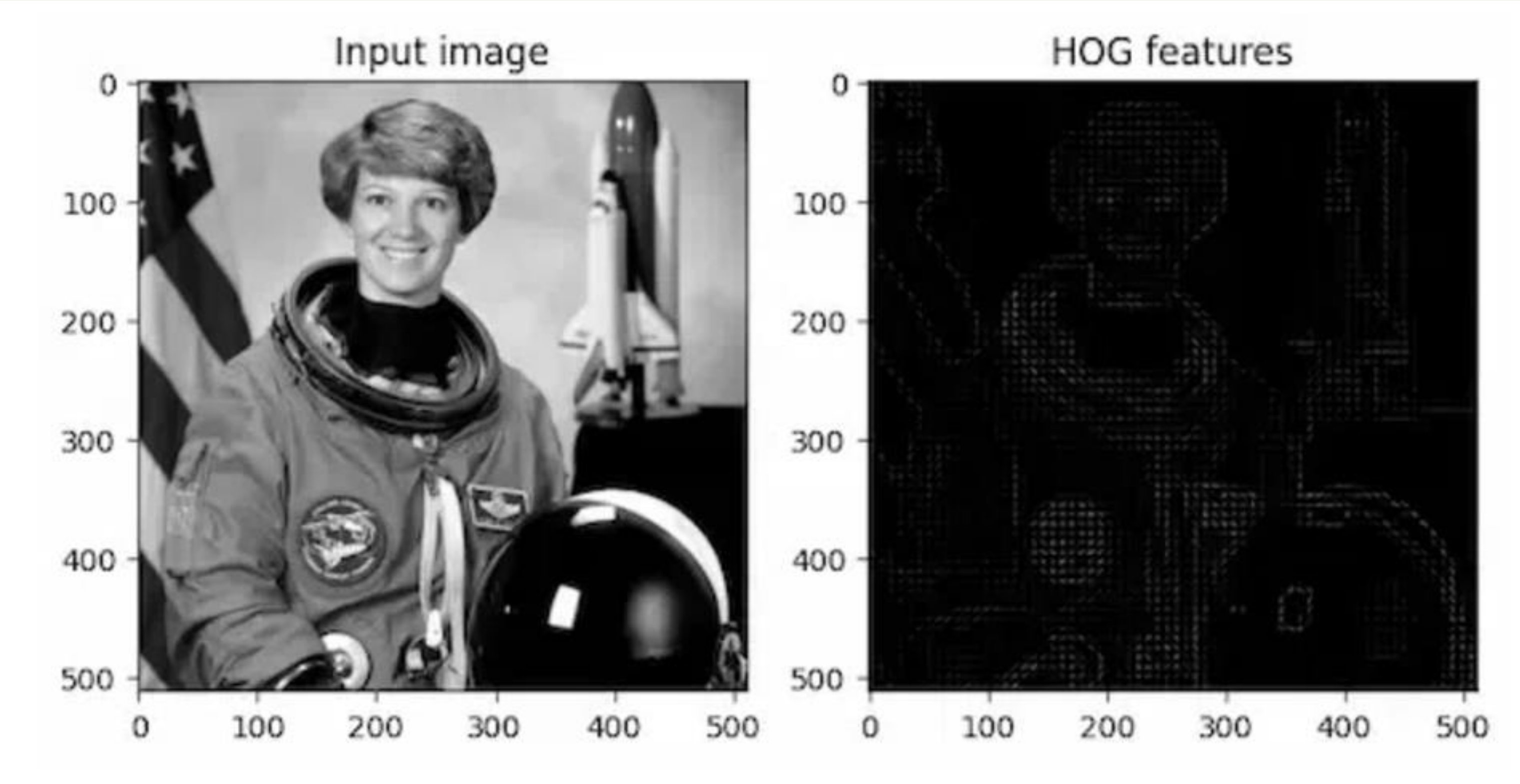
- HOG features were first introduced by Dalal and Triggs in 2005 as a robust feature extraction method for pedestrian detection.
- The core idea behind HOG is to capture the distribution of gradient orientations in an image, which can be used to describe the shape and appearance of objects.
- HOG features are computed by dividing an image into small cells, calculating the gradient orientations within each cell, and then aggregating these orientations into a histogram. This histogram represents the distribution of gradient orientations, which can be used as a feature vector for object detection.

Advantages of HOG Feature

- HOG features have several benefits that make them an attractive choice for object detection:
  - *Robustness to lighting changes: HOG features are invariant to changes in lighting conditions, making them suitable for object detection in real-world scenarios.*
  - *Robustness to occlusions: HOG features can handle partial occlusions, allowing for accurate object detection even when objects are partially hidden.*
  - *Computational efficiency: HOG features can be computed efficiently, making them suitable for real-time object detection applications.*
  - *Flexibility: HOG features can be used with various classification algorithms, such as Support Vector Machines (SVMs) and Random Forests, to name a few.*

HOG Example with Python

- By computing the distribution of local intensity gradients or edge directions in an image, HOG features capture the presence of specific shapes and edges.



HOG Example with Skimage

- Added a title to the figure for better context.
- Added annotations to the subplots for clarity.
- Ensure compatibility with color images.
- Removed the unnecessary channel axis specification.
- Improved layout and spacing for better aesthetics.

Neural Network-Based Techniques

- With the advent of deep learning, neural network-based techniques have become the standard for object detection. These methods include:

  - *Convolutional Neural Networks (CNNs): CNNs are widely used for object detection due to their ability to automatically learn features from data. They are the backbone of many state-of-the-art object detection models.*
  - *Region-Based CNN (R-CNN): This method generates region proposals and then classifies each region using a CNN. Variants like Fast R-CNN and Faster R-CNN have improved the speed and accuracy of this approach.*

Neural Network-Based Techniques

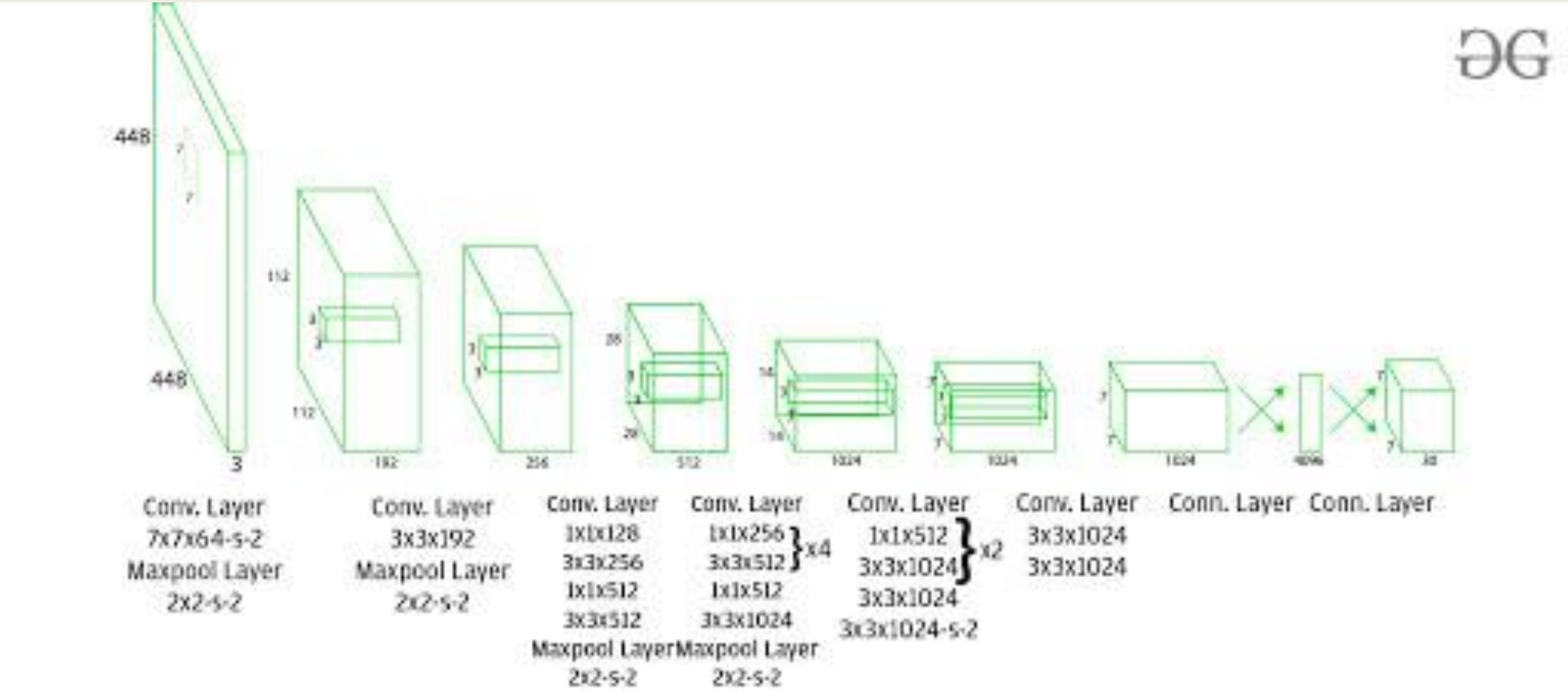
- *You Only Look Once (YOLO): YOLO is a single-stage object detector that divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell in one pass, making it extremely fast.*

- *Single Shot MultiBox Detector (SSD): SSD is another single-stage detector that uses a series of convolutional layers to predict bounding boxes and class scores for multiple objects in an image.*

YOLO : You Only Look Once – Real Time Object Detection

- YOLO was proposed by Joseph Redmond et al. in 2015. It was proposed to deal with the problems faced by the object recognition models at that time,
- Fast R-CNN is one of the state-of-the-art models at that time but it has its own challenges such as this network cannot be used in real-time, because it takes 2-3 seconds to predicts an image and therefore cannot be used in real-time. Whereas, in YOLO we have to look only once in the network i.e. only one forward pass is required through the network to make the final predictions.

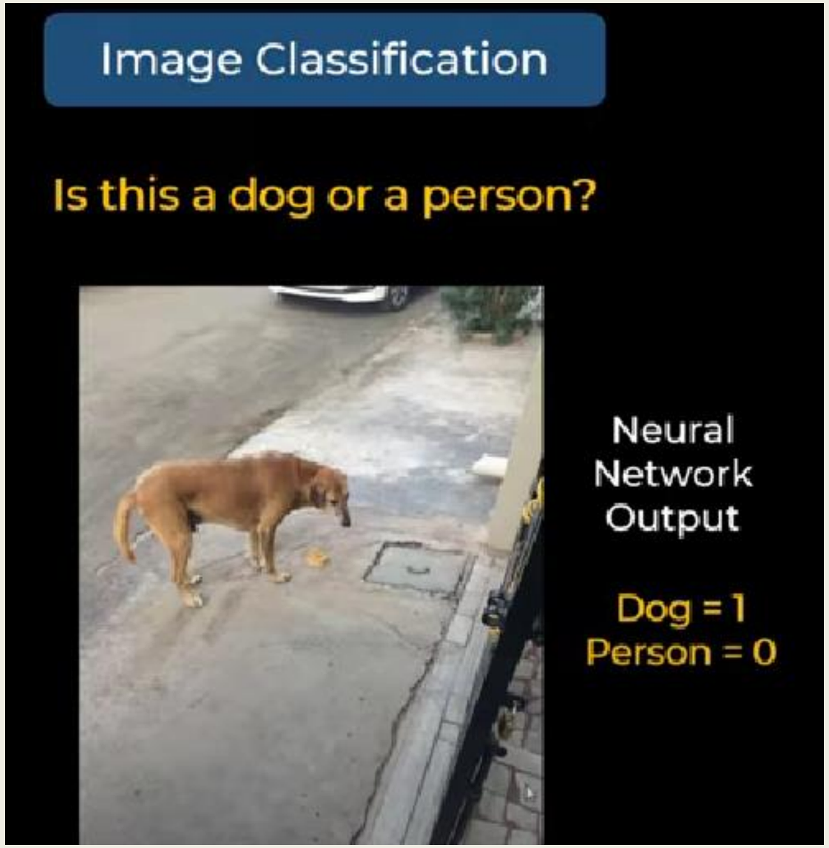
Architecture

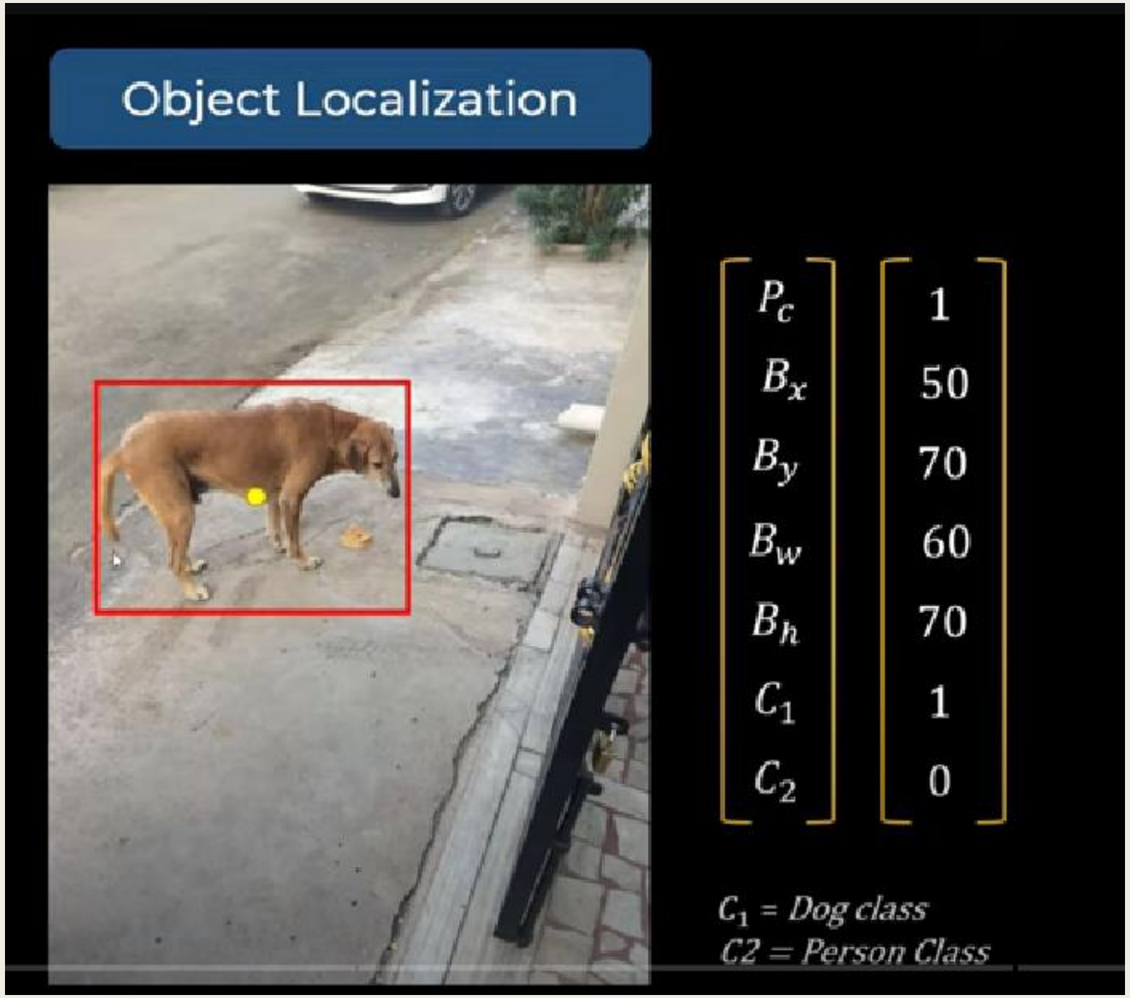


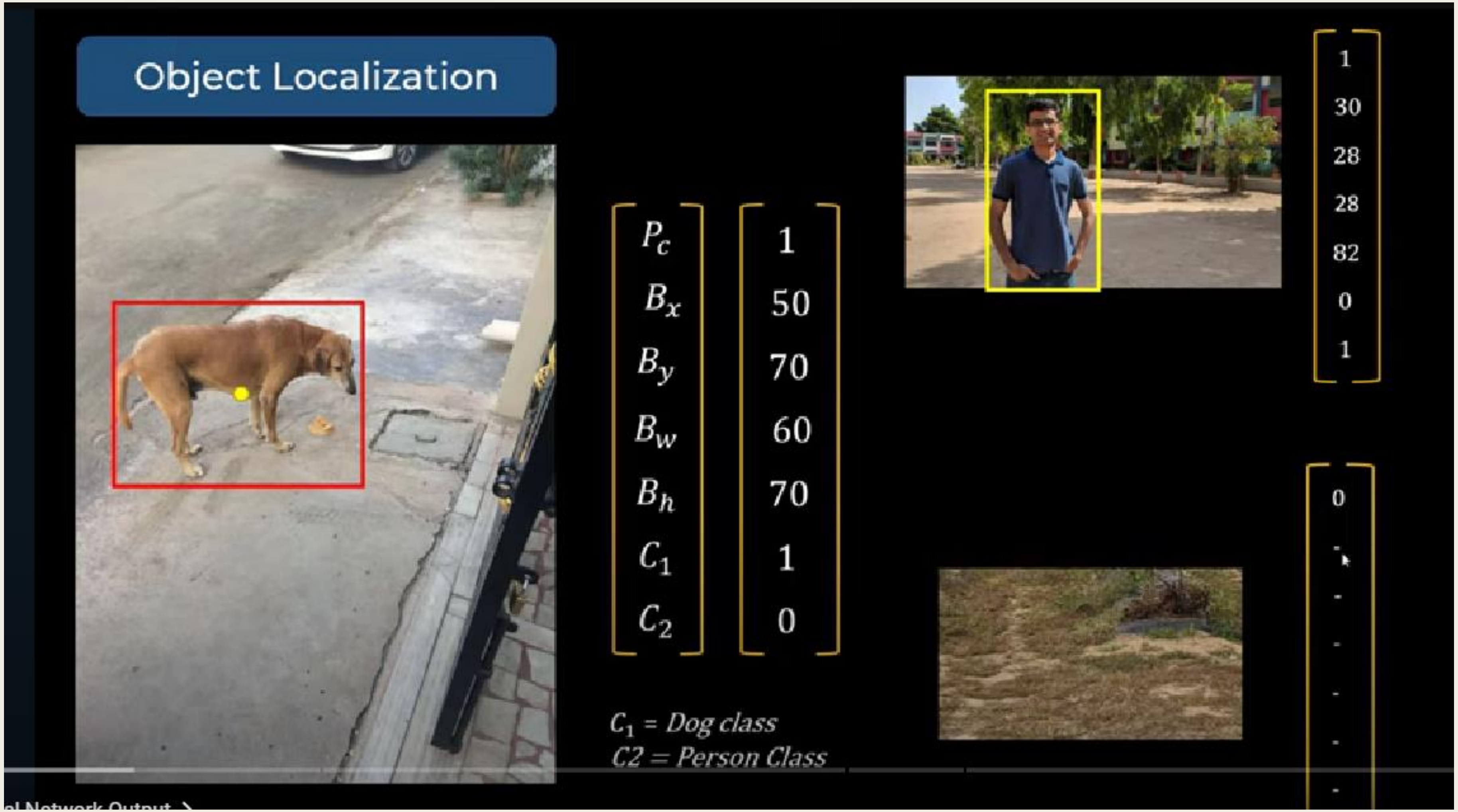
Architecture

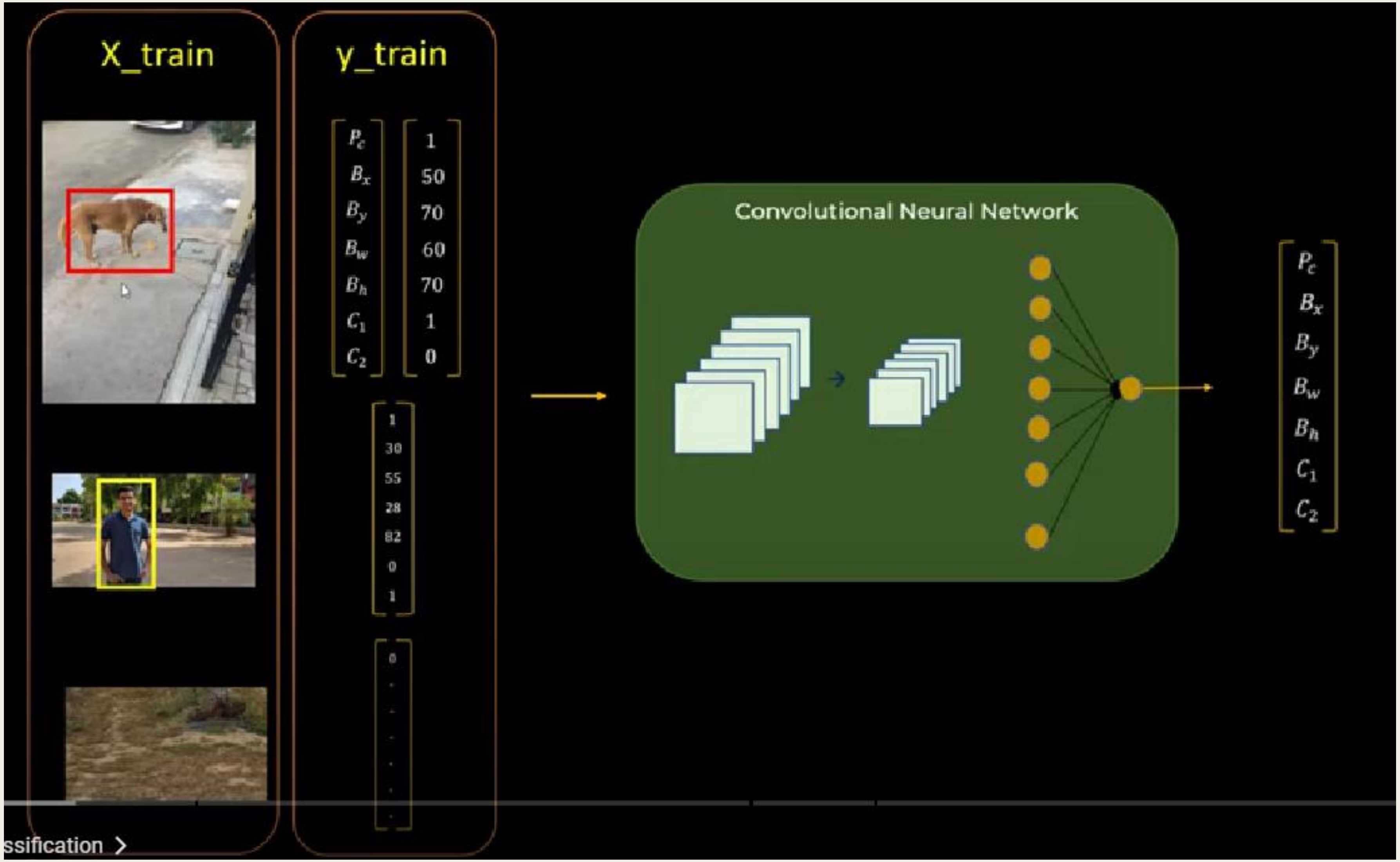
- This architecture takes an image as input and resizes it to *448*448* by keeping the aspect ratio same and performing padding. This image is then passed in the CNN network.
- This model has *24 convolution layers*, *4 max-pooling layers followed by 2 fully connected layers*.
- For the reduction of the number of layers (Channels), we use *1*1* convolution that is followed by *3*3* convolution. Notice that the last layer of YOLOv1 predicts a cuboidal output.
- This is done by generating *(1, 1470)* from final fully connected layer and reshaping it to size *(7, 7, 30)*.
- This architecture uses Leaky ReLU as its activation function in whole architecture except the last layer where it uses linear activation function.

Example -- YOLO



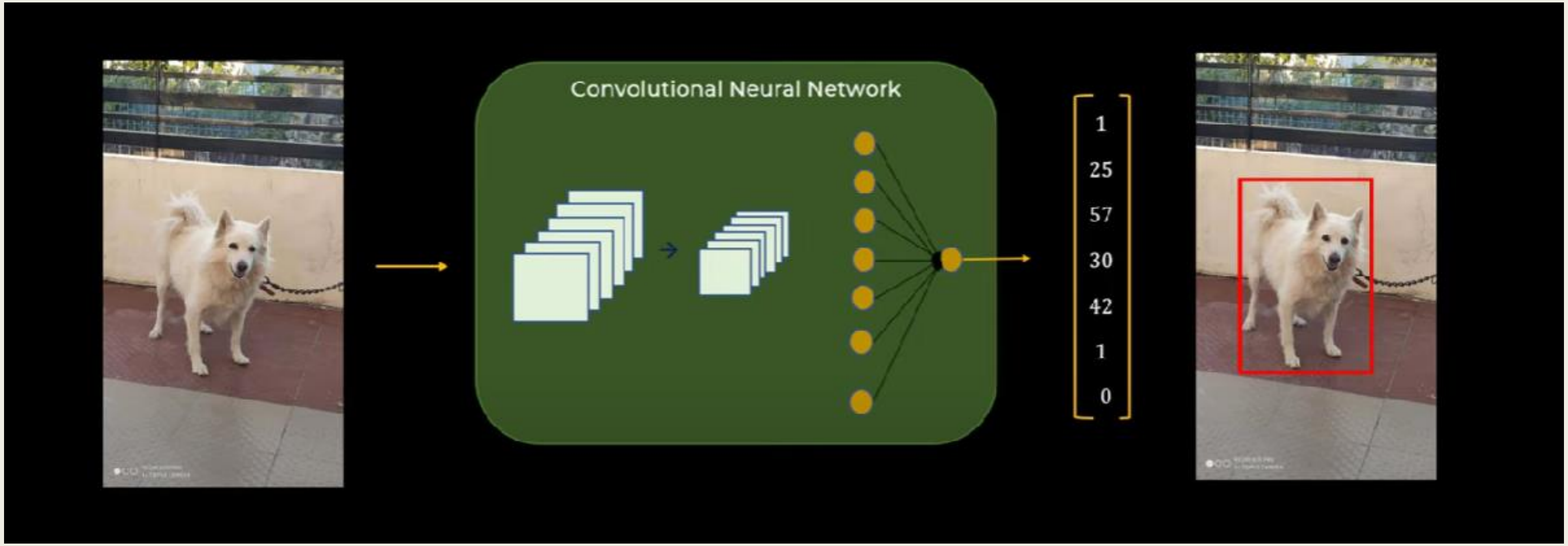


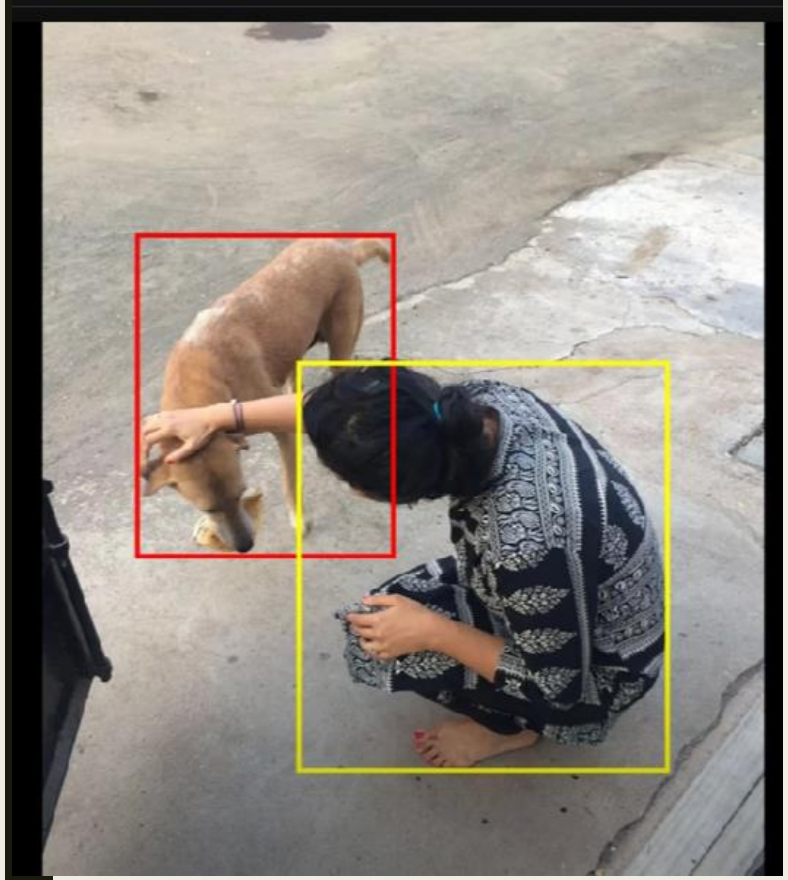

Object Localization
P_c 1
B_x 50
B_y 70
B_w 60
B_h 70
C_1 1
C_2 0
C_1 = Dog class
C2 = Person Class
1
30
28
28
82
0
1
0

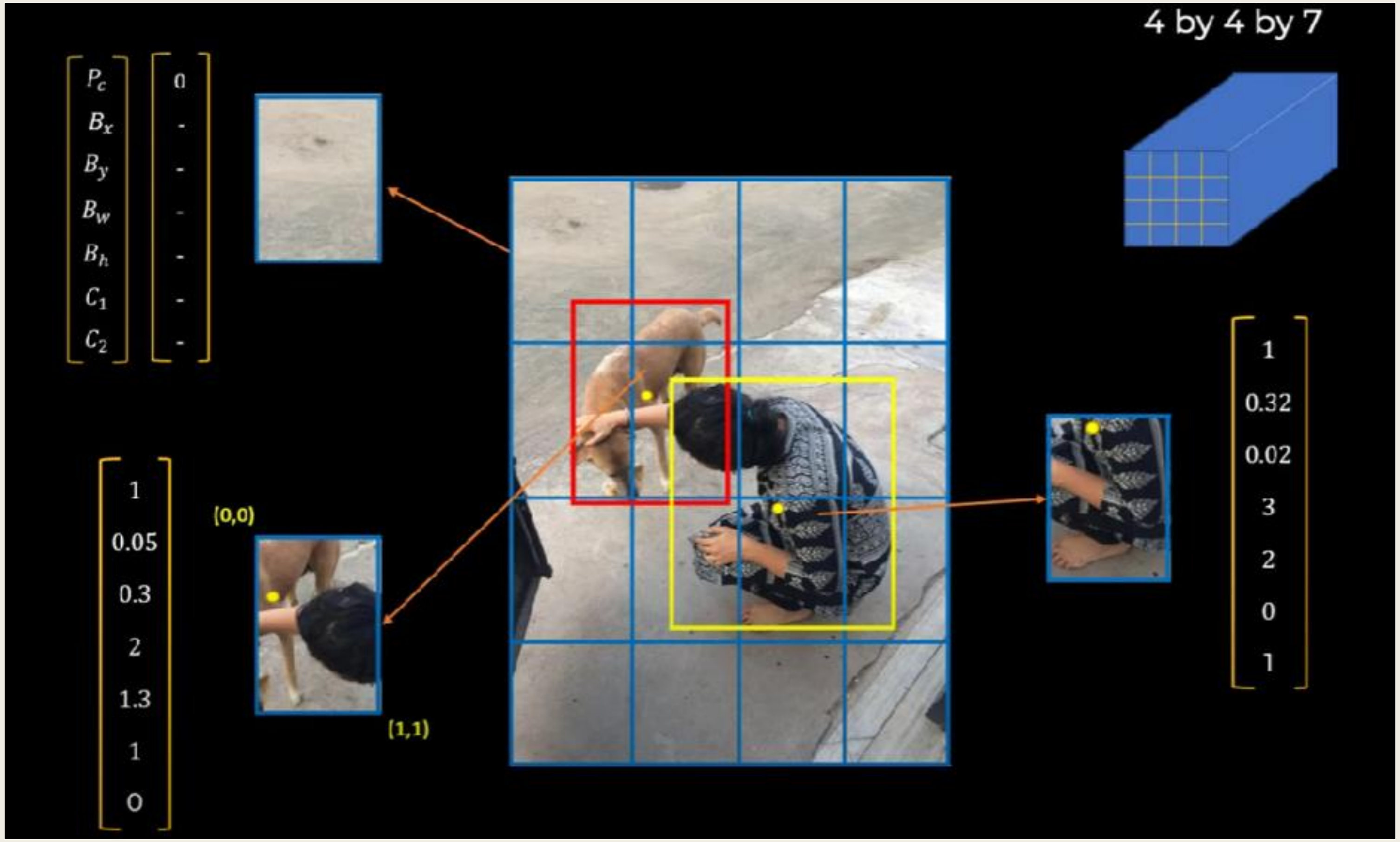


X_train
y_train
P_c B_x B_y B_w B_h C_1 C_2
1 50 70 60 70 1 0
1 30 55 28 82 0 1
0
Convolutional Neural Network
P_c B_x B_y B_w B_h C_1 C_2
ssification

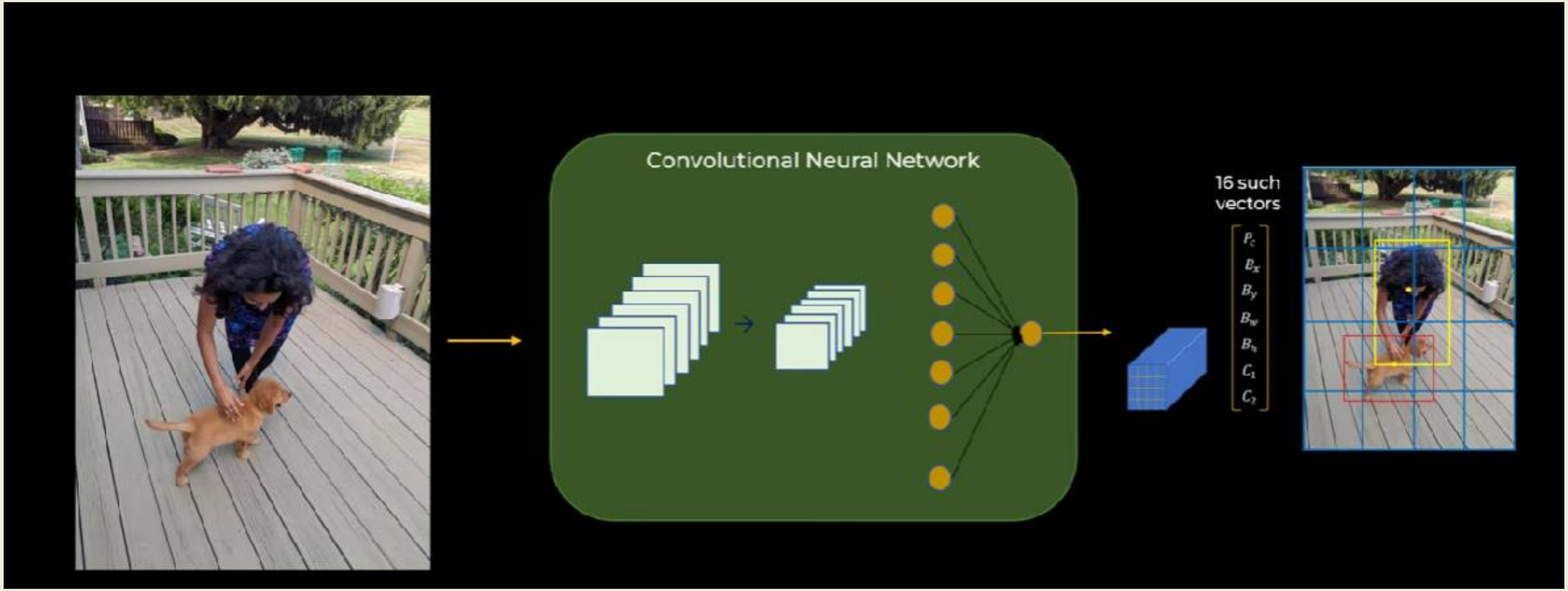
Prediction





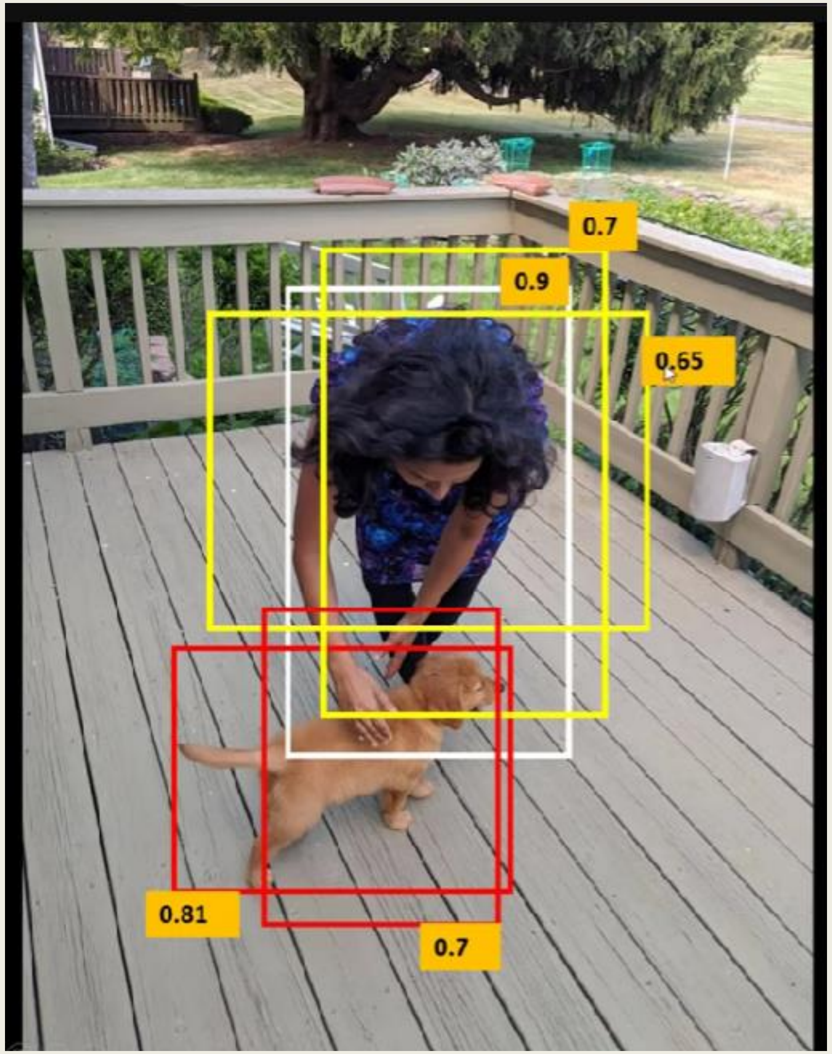

4 by 4 by 7
P_c
B_x
B_y
B_w
B_h
C_1
C_2
0
-
-
-
-
-
-
1
0.05
0.3
2
1.3
1
0
(0,0)
(1,1)
1
0.32
0.02
3
2
0
1

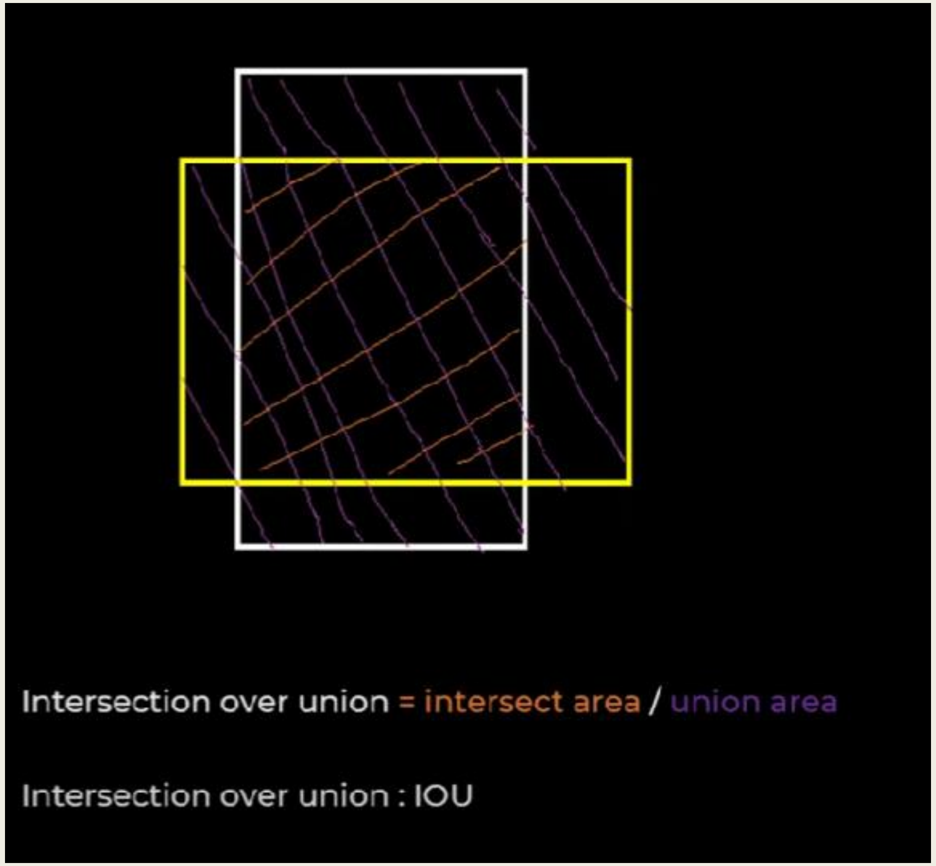
Prediction



Convolutional Neural Network
16 such vectors
P_c
B_x
B_y
B_w
B_h
C_1
C_2

Challenge of YOLO





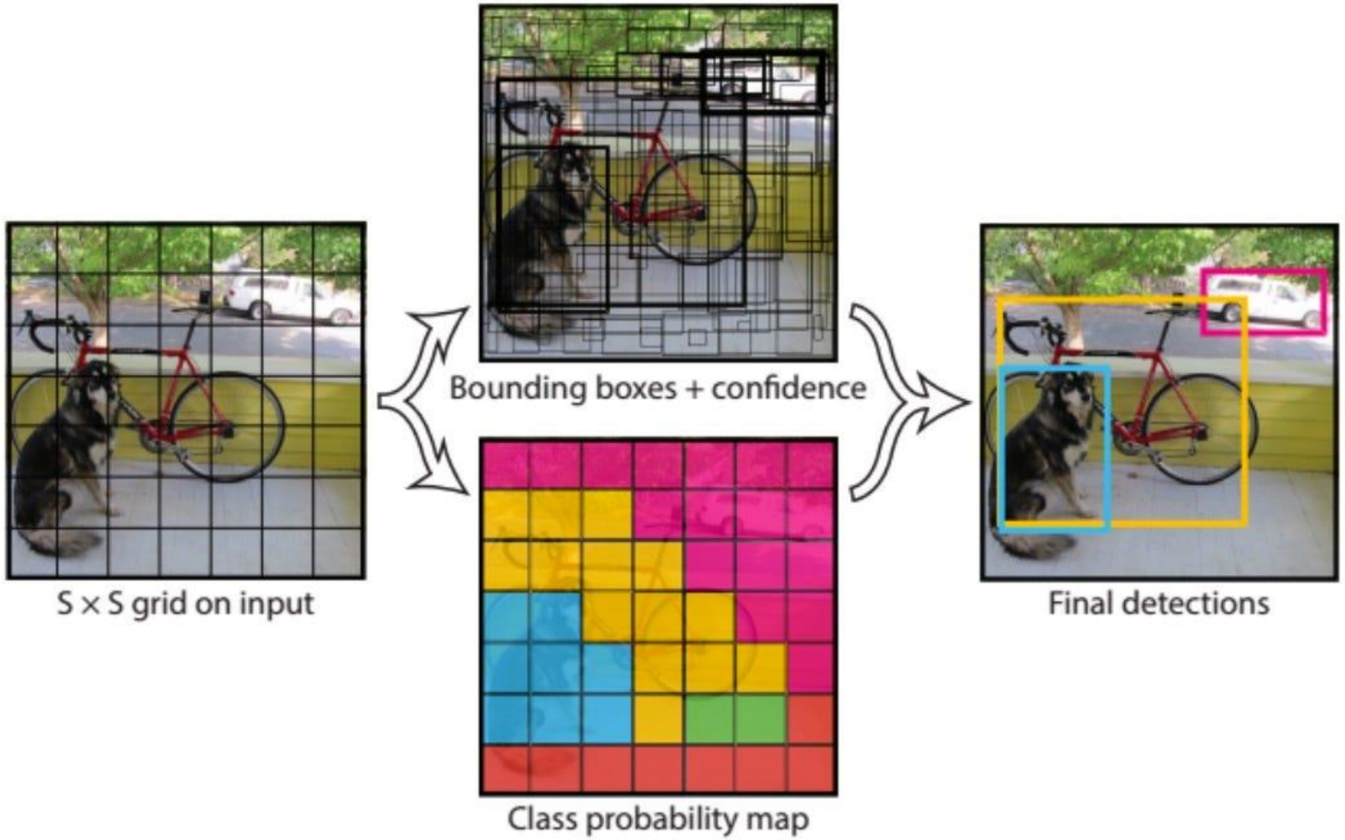


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

References

- https://www.geeksforgeeks.org/introduction-to-object-detection-using-image-processing/
- https://www.geeksforgeeks.org/hog-feature-visualization-in-python-using-skimage/
- https://www.geeksforgeeks.org/yolo-you-only-look-once-real-time-object-detection/
- Video Tutorial for YOLO
  - https://www.youtube.com/watch?v=ag3DLKsl2vk