# OBJECT DETECTION


CLUSTER INNOVATION CENTRE
UNIVERSITY OF DELHI

Sahiljit Sandhu (11833)
Ritik Arya (11829)

May 2021

**ACKNOWLEDGEMENT**

With a deep sense of gratitude, we express my dearest indebtedness to **Prof. Anu Yadav**, for her support throughout the duration of our project. We would like to thank her for giving us the opportunity to work on this wonderful project. Her learned advice and constant encouragement has helped us to complete this project. It is a privilege for us to be her students.

# ABSTRACT

## OBJECT DETECTION

by

Sahiljit Sandhu
Ritik Arya

CLUSTER INNOVATION CENTER, 2021

Computer Vision is the branch of the science of computers and software systems which can recognize as well as understand images and scenes. Computer Vision consists of various aspects such as image recognition, object detection, image generation, image super-resolution and many more. The applications of object detection include tracking objects, video surveillance, pedestrian detection, people counting, self-driving cars, face detection, ball tracking in sports and many more. In this project, we are using highly accurate object detection-algorithms and methods such as R-CNN, Fast-RCNN, Faster-RCNN and fast yet highly accurate ones like YOLO. Using these methods and algorithms, based on deep learning we can detect each and every object in image by the area object in an highlighted rectangular box and identify each and every object and assign its tag to the object.

# TABLE OF CONTENTS

# 1. INTRODUCTION

Object Detection is the process of finding and recognizing real-world object instances such as cars, bikes, TV, flowers, and humans out of images or videos. An object detection technique lets you understand the details of an image or a video as it allows for the recognition, localization, and detection of multiple objects within an image. Humans can easily detect and identify objects present in an image because the human visual system is fast and accurate and can perform complex tasks like identifying multiple objects with little conscious thought. And with the rapid development in deep learning, more powerful tools, which are able to learn semantic, high-level, deeper features and with the availability of large amounts of data, faster GPUs, and better algorithms, we can now easily train computers to detect and classify multiple objects within an image with high accuracy. Object detection is a vast, vibrant and complex area of computer vision. It further includes two tasks i.e image classification and object localization. Where, in **image classification** we try to predict the type or class of an object in an image.

For image classification : -

- Input: An image with a single object, such as a photograph.
- Output: A class label (e.g. one or more integers that are mapped to class labels).

And in **object Localization**, we try to locate the presence of objects in an image and indicate their location with a bounding box.

For Object Localization : -

- Input: An image with one or more objects, such as a photograph.
- Output: One or more bounding boxes (e.g. defined by a point, width, and height)

And both the above tasks add up to the task of object detection where we try to locate the presence of objects with a bounding box and types or classes of the located objects in an image.

# 2. CHALLENGES

Working on complex problems like object detection presents us with many challenges which we will discuss in this section.

## 2.1 Localization of the object

In object detection we not only want to classify image objects but also want to determine the objects' positions, generally referred to as the object localization task.

## 2.2 Computation speed

Object detection algorithms need to accurately classify and localize important objects as well as they also need to be incredibly fast at prediction time to meet the real-time demands of video processing.

## 2.3 Multiple spatial scales and aspect ratios

For many applications of object detection, items of interest may appear in a wide range of sizes and aspect ratios. As we feed data to our model, the input images could be of different sizes as well as the objects inside the images could also be of different aspect ratios.

# 3. OBJECTIVE

The objective of our project is to discuss, compare and implement different deep learning models that solves the problem of object detection. We discuss frameworks which handle different sub-problems, such as occlusion, clutter and low resolution, with different degrees of modifications on R-CNN. Generic object detection pipelines which provide base architectures for other related tasks is also covered in detail.

# 4. METHODOLOGY

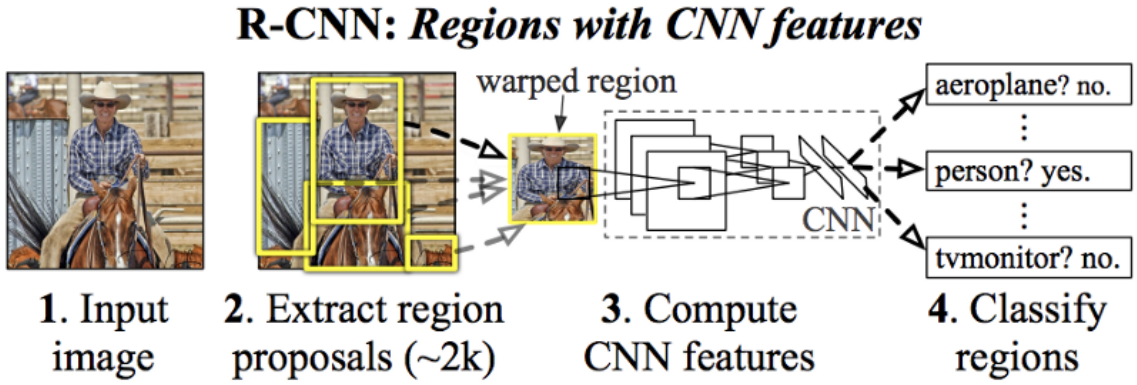
## 3.1 R-CNN

The R-CNN was described in the 2014 paper by Ross Girshick, et al. from UC Berkeley titled "Rich feature hierarchies for accurate object detection and semantic segmentation." It is one of the first large and successful applications of convolutional neural networks to solve the problem of object localization, detection, and segmentation. The approach was demonstrated on benchmark datasets, achieving then state-of-the-art results on the VOC-2012 dataset and the 200-class ILSVRC-2013 object detection dataset.

In R-CNN, we use selective search to extract regions from the image we think that an object is present in. **Selective search** algorithm works as follows:-

1. Generate initial sub-segmentation, we generate many candidate regions.
2. Use greedy algorithms to recursively combine similar regions into larger ones.
3. Use the generated regions to produce the final candidate region proposals.

The pipeline of R-CNN is shown in the following image:-




These candidate region proposals are warped into a square, scaled using OpenCV and then fed into a convolutional neural network that produces a dimensional feature vector as output. The CNN acts as a feature extractor and the output dense layer consists of the

features extracted from the image and the extracted features are fed into an SVM to classify the presence of the object within that candidate region proposal. In addition to predicting the presence of an object within the region proposals, the algorithm also predicts four values which are offset values to increase the precision of the bounding box.

Problems with R-CNN:-

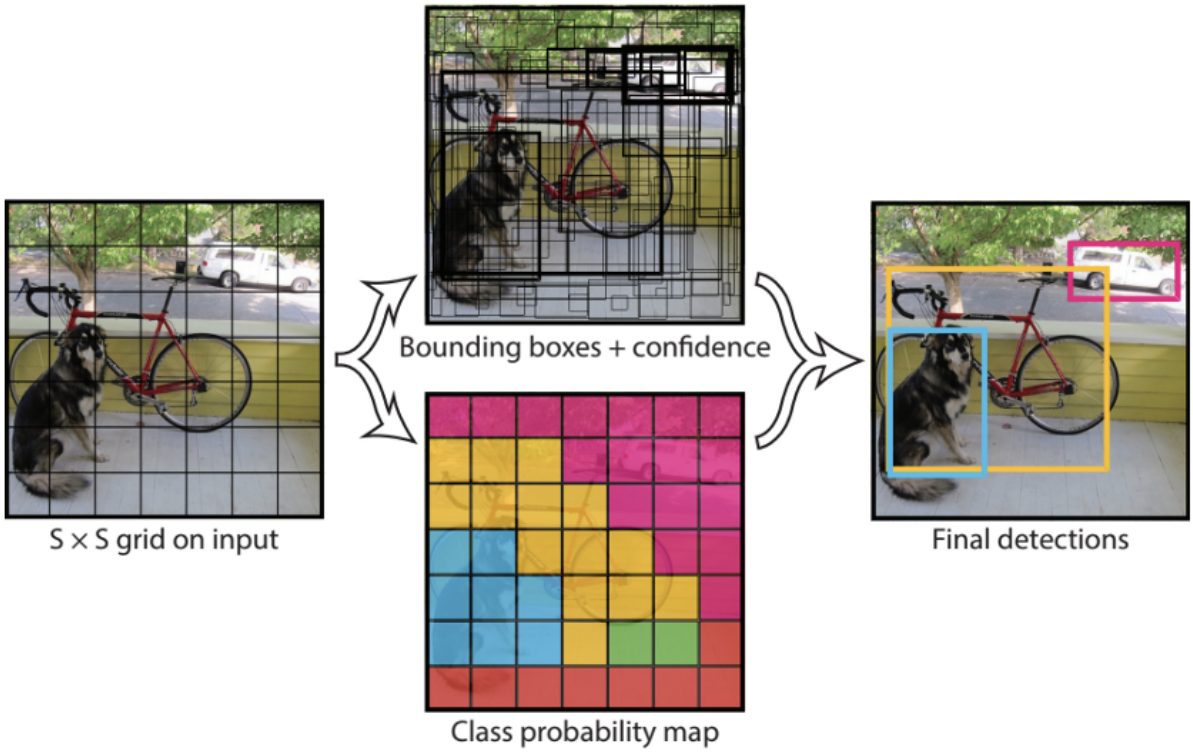
- It still takes a huge amount of time to train the network as you would have to classify 2000 region proposals per image.
- It cannot be implemented real time as it takes around 47 seconds for each test image.
- The selective search algorithm is a fixed algorithm. Therefore, no learning is happening at that stage. This could lead to the generation of bad candidate region proposals.

### 3.2 YOLO - You Only Look Once

All of the previous object detection algorithms use regions to localize the object within the image. The network does not look at the complete image. Instead, parts of the image which have high probabilities of containing the object. YOLO or You Only Look Once is an object detection algorithm much different from the region based algorithms seen above. In YOLO a single convolutional network predicts the bounding boxes and the class probabilities for these boxes.

How YOLO works is that we take an image and split it into an SxS grid, within each of the grid we take m bounding boxes. For each of the bounding box, the network outputs a class probability and offset values for the bounding box. The bounding boxes having the

class probability above a threshold value is selected and used to locate the object within the image.




YOLO is orders of magnitude faster(45 frames per second) than other object detection algorithms. The limitation of the YOLO algorithm is that it struggles with small objects within the image, for example it might have difficulties in detecting a flock of birds. This is due to the spatial constraints of the algorithm.

**Loss Function of Yolo**

YOLO predicts multiple bounding boxes per grid cell. To compute the loss for the true positive, we only want one of them to be responsible for the object. For this purpose, we

select the one with the highest IoU (intersection over union) with the ground truth. This strategy leads to specialization among the bounding box predictions. Each prediction gets better at predicting certain sizes and aspect ratios.

YOLO uses sum-squared error between the predictions and the ground truth to calculate loss. The loss function comprises of:

- the classification loss.
- the localization loss (errors between the predicted boundary box and the ground truth).
- the confidence loss (the objectness of the box).

The final loss looks like :-

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left( C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left( C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left( p_i(c) - \hat{p}_i(c) \right)^2$$

# 5. SYSTEM REQUIREMENTS

Following are the important libraries and dependencies that are required for this project :-

## 5.1 Tensorflow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

## 5.2 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products.

*pip install opencv-python -command*

## 5.3 Pillow

It is a  free Python programming language library that provides support to open, edit and save several different formats of image files. Windows, Mac OS X and Linux are available for this.

*pip install pillow -command*

## 5.4 Numpy

NumPy is a library of Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate over these arrays.

*pip install numpy -command*

# 6. RESULTS

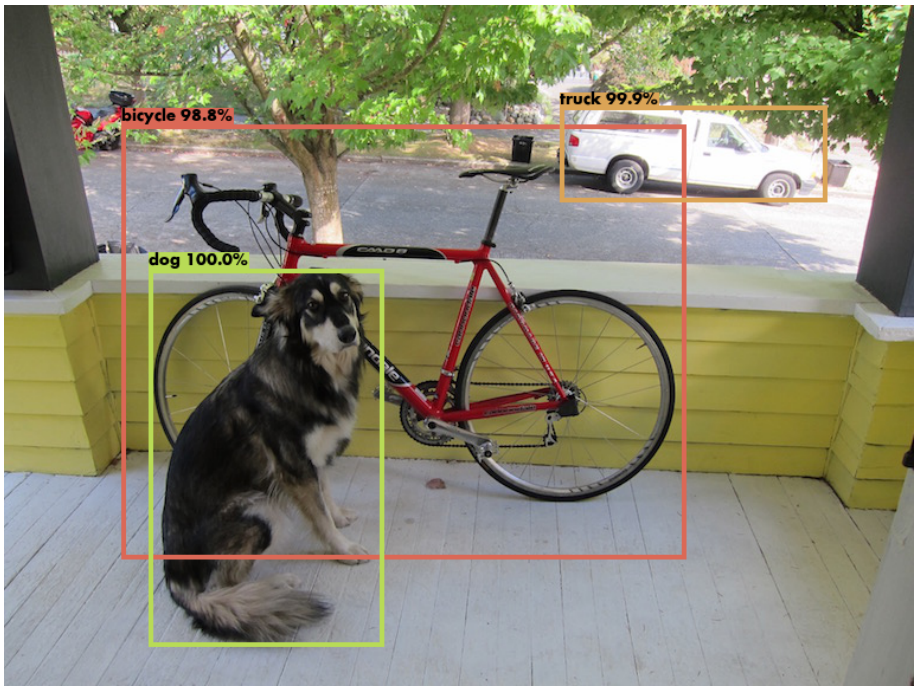
Following are the results after running the YOLO model on sample images.

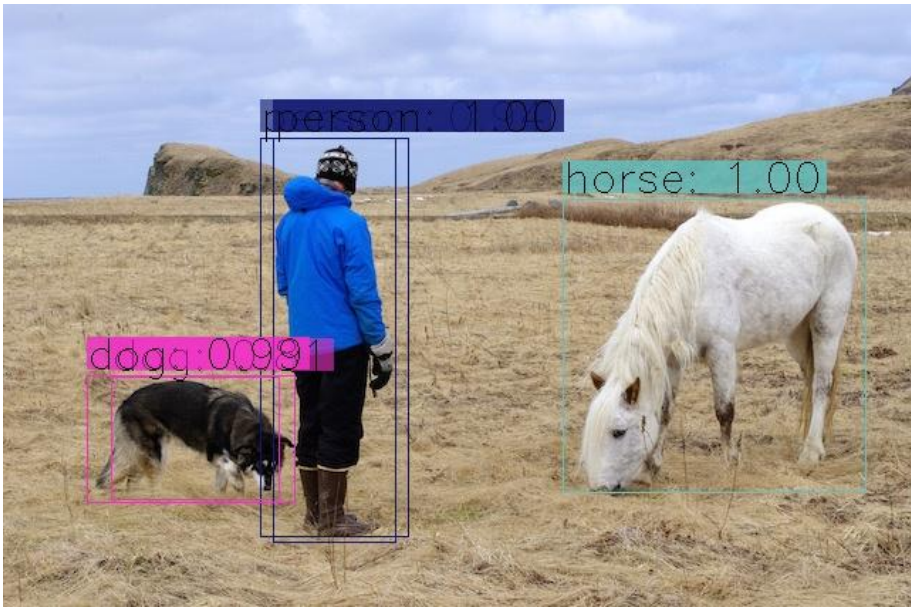
**Sample images**

**1.**

2.

NOTE: - Testing the model with IoU (Intersection over Union ratio used in non-max suppression) threshold and confidence threshold both set to 0.5.



# 7. CONCLUSION

We now have a better understanding of how we can localize objects while classifying them in an image. We also learned to combine the concept of classification and localization with the convolutional implementation of the sliding window to build an object detection system. By using the above deep learning models and based on experimental results we are able to detect objects more precisely and identify the objects individually with the exact location of an object in the picture in the x,y axis.