**Het Bhatt(D003)**

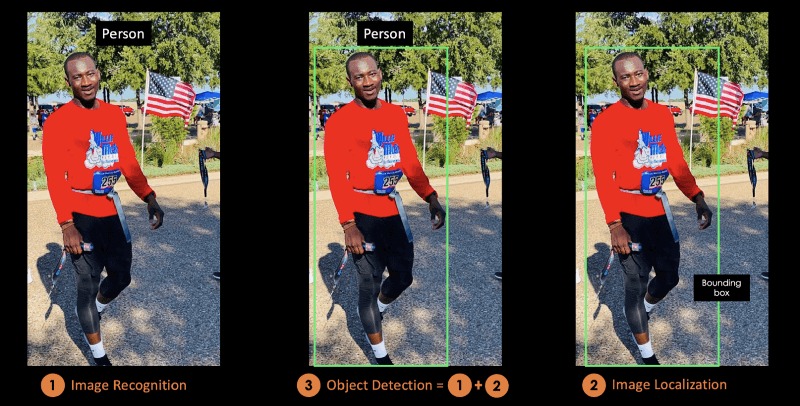
**Introduction to Object Detection**

Object detection is a technique used in computer vision for the identification and localization of objects within an image or a video.

Image Localization is the process of identifying the correct location of one or multiple objects using bounding boxes, which correspond to rectangular shapes around the objects.

This process is sometimes confused with image classification or image recognition, which aims to predict the class of an image or an object within an image into one of the categories or classes.

The illustration below corresponds to the visual representation of the previous explanation. The object detected within the image is “Person.”



In this conceptual blog, you will first understand the benefits of object detection, before introducing YOLO, the state-of-the-art object detection algorithm.

In the second part, we will focus more on the YOLO algorithm and how it works. After that, we will provide some real-life applications using YOLO.

The last section will explain how YOLO evolved from 2015 to 2020 before concluding on the next steps.

**What is YOLO?**

In their well-known academic work "You Only Look Once: Unified, Real-Time Object Identification," Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi unveiled the cutting-edge real-time object detection technique known as You Only Look Once (YOLO) in 2015.

The authors spatially separate bounding boxes and use a single convolutional neural network to assign probabilities to each of the detected images, framing the object identification problem as a regression problem rather than a classification task (CNN).

You will be able to construct Keras-based deep neural networks for image classification applications by enrolling in the Image Processing with Keras in Python course.

Convolutional neural networks and how to use them to create models with far greater capability are covered in depth learning with pytorch, if that is more your style.

## What Makes YOLO Popular for Object Detection?

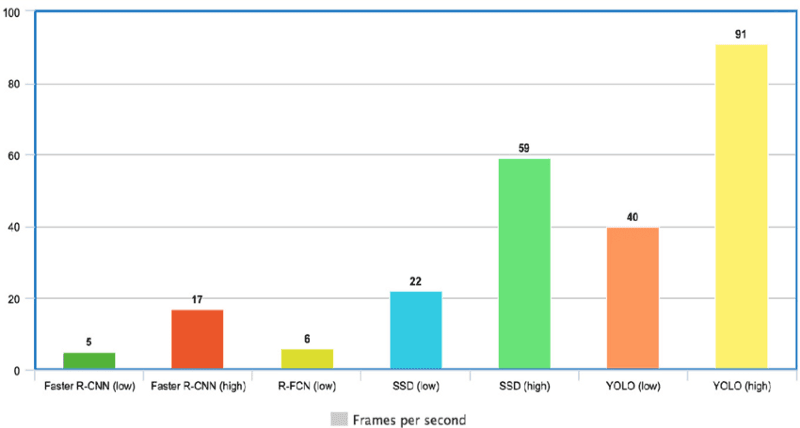
Some of the reasons why YOLO is leading the competition include its:

* Speed
* Detection accuracy
* Good generalization
* Open-source

### 1- Speed

Because it doesn't deal with complicated pipelines, YOLO is incredibly quick. At 45 frames per second, it can process images (FPS). YOLO is a fantastic choice for real-time processing since it can achieve more than twice the mean Average Precision (mAP) compared to other real-time systems.

With 91 FPS, YOLO is clearly superior than the other object detectors, as shown in the graph below..



YOLO Speed compared to other state-of-the-art object detectors ([**source**](https://www.researchgate.net/figure/Comparison-of-frames-processed-per-second-FPS-implementing-the-Faster-R-CNN-R-FCN-SSD_fig6_342570032))

### 2- High detection accuracy

### With a very low amount of background mistakes, YOLO's accuracy greatly outpaces that of other cutting-edge models.

### 3- Better generalization

This is particularly true for the updated YOLO versions, which will be covered in more detail later on in the text. With those improvements, YOLO went a little further and offered improved generalisation for new domains, making it ideal for applications that require quick and reliable object identification.

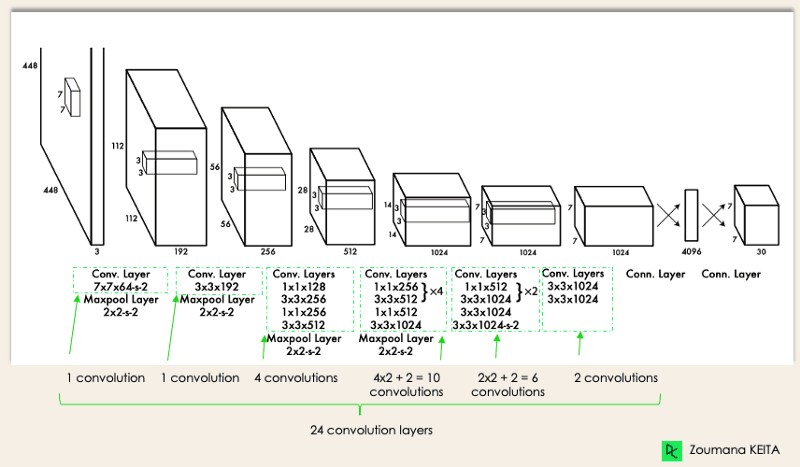
The Automatic Detection of Melanoma with Yolo Deep Convolutional Neural Networks article, for instance, demonstrates that, when compared to YOLOv2 and YOLOv3, the first version of YOLOv1 has the lowest mean average precision for the automatic detection of melanoma disease.

### 4- Open source

Making YOLO open-source encouraged the neighbourhood to continuously enhance the model. This is among the factors that have contributed to YOLO's rapid rise.

## **YOLO Architecture**

The Google architecture of YOLO is comparable. It has a total of 24 convolutional layers, four max-pooling layers, and two fully connected layers, as shown below.



YOLO Architecture from the[**original paper**](https://arxiv.org/pdf/1506.02640.pdf) (Modified by Author)

The architecture works as follows:

## • Before passing the input image through the convolutional network, it is resized to 448x448.

## • To create a cuboidal output, a 3x3 convolution is used after a 1x1 convolution to reduce the number of channels.

## • Some extra techniques, such as batch normalisation and dropout, respectively regularise the model and prevent it from overfitting. • The activation function used internally is ReLU, with the exception of the final layer, which employs a linear activation function.

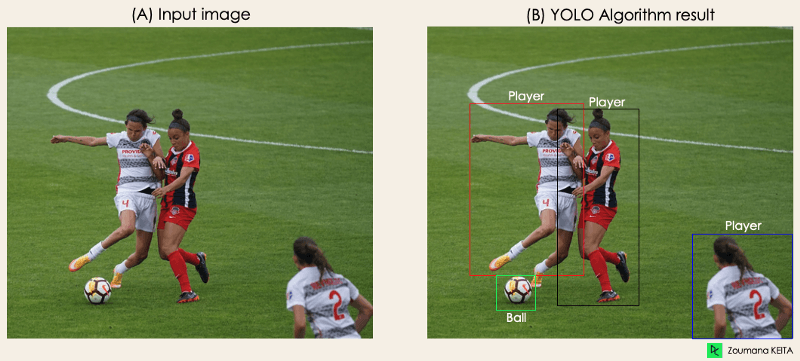
## You will be prepared to utilise Keras to train and test intricate, multi-output networks and further your understanding of deep learning once you have finished the Deep Learning in Python course.

## **How Does YOLO Object Detection Work?**

Now that you are familiar with the architecture, let's take a high-level look at how the YOLO algorithm handles object detection in the context of a straightforward use case.

"Assume you created a YOLO application that can identify soccer balls and players from a given photograph.

But how can you describe this procedure to someone, particularly someone who is not familiar with it? That is the only purpose of this paragraph. You will comprehend the entire object detection technique used by YOLO, including how to obtain image (B) from image (A).

Image by Author

The algorithm works based on the following four approaches:

* Residual blocks
* Bounding box regression
* Intersection Over Unions or IOU for short
* Non-Maximum Suppression.

Let’s have a closer look at each one of them.

### 1- Residual blocks

This first step starts by dividing the original image (A) into NxN grid cells of equal shape, where N in our case is 4 shown on the image on the right. Each cell in the grid is responsible for localizing and predicting the class of the object that it covers, along with the probability/confidence value.

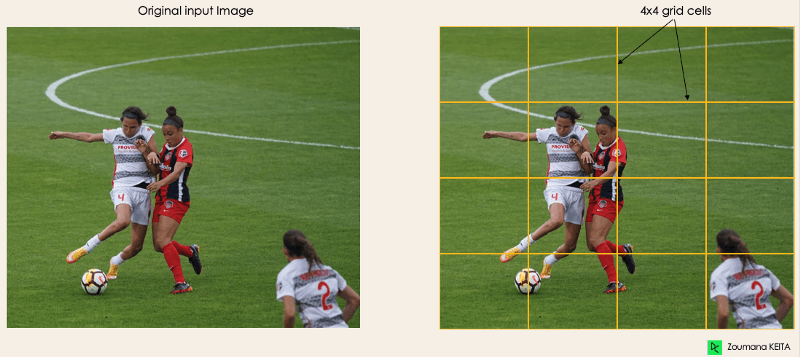


Image by Author

### 2- Bounding box regression

The next step is to identify the bounding boxes, which are rectangles that highlight all of the image's objects. As many bounding boxes as there are objects in a given image is possible.

The properties of these bounding boxes are determined by YOLO using a single regression module, where Y is the final vector representation of each bounding box.

Y = [bx, by, bh, bw, c1]

This is particularly crucial during the model's training phase.

• The probability score of the grid containing an object is represented by pc. For example, every red grid will have a probability value greater than zero. The simplified form is shown on the right since there is no chance that any of the cells will be yellow (insignificant).

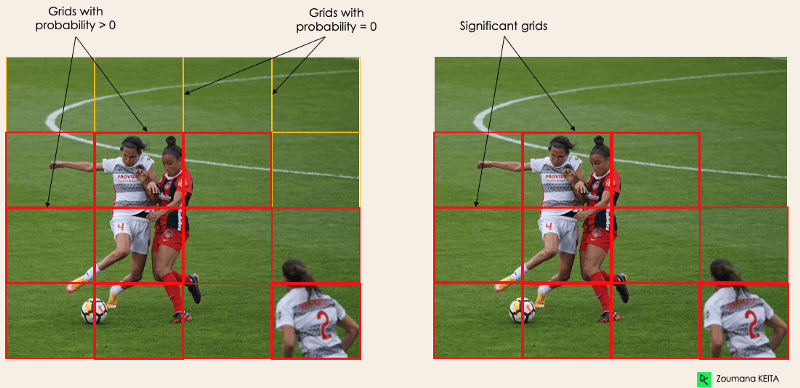


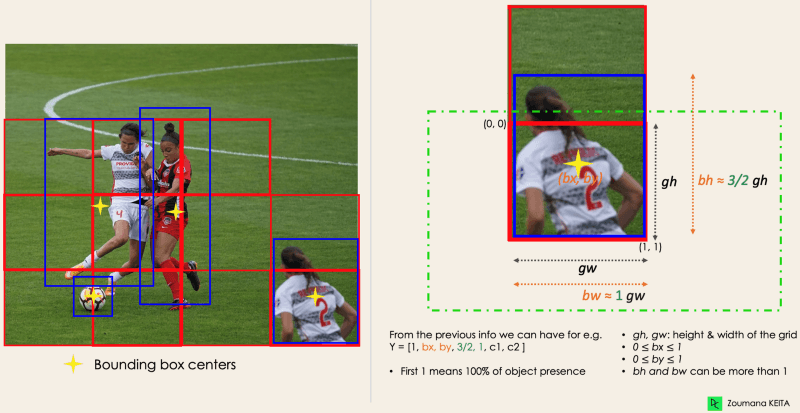
Image by Author

• The bounding box's center's x and y coordinates in relation to the surrounding grid cell are given by bx, by.

• bh, bw stand for the bounding box's height and breadth in relation to the surrounding grid cell.

• The two classes Player and Ball are represented by c1 and c2. If your use case calls for more classes, we can have them.

To understand, let’s pay closer attention to the player on the bottom right.

Image by Author

### 3- Intersection Over Unions or IOU

Despite not all of them being significant, a single object in an image might frequently have many grid box possibilities for prediction. The IOU's (a value between 0 and 1) purpose is to eliminate such grid boxes and retain only the necessary ones. Here is the reasoning for it:

• The IOU selection threshold is set by the user and can be, for example, 0.5.

• After that, YOLO determines each grid cell's IOU, which is calculated by dividing the intersection area by the union area.

• Finally, it evaluates grid cells with an IOU > threshold rather than those predicted to have an IOU threshold.

An example of using the grid selection method on the object in the bottom left is shown below. We can see that the object had two potential grid candidates at first, but only "Grid 2" was ultimately chosen.



Image by Author

### 4- Non-Max Suppression or NMS

Setting an IOU threshold is not always sufficient because an item may contain numerous boxes with IOU that exceeds the threshold, and leaving all of those boxes open could result in the inclusion of noise. Here, NMS can be used to keep only the boxes with the highest likelihood of being discovered.

## YOLO Applications

### YOLO object detection has a variety of uses in daily life. Some of them in the fields of healthcare, agriculture, security monitoring, and self-driving automobiles will be discussed in this part.

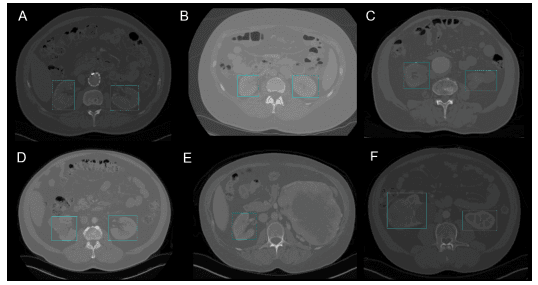
### **Application in industries**

Many useful areas, including healthcare and agriculture, have adopted object detection. Let's use concrete examples to clarify each one.

Healthcare

Due to biological differences between patients, localising organs in real time during surgery might be difficult. Using YOLOv3, Kidney Recognition in CT was able to more easily locate kidneys in CT scans in both 2D and 3D.

You can master the principles of examining, modifying, and measuring biomedical image data using Python by taking the course Biomedical Image Analysis in Python.



2D Kidney detection by YOLOv3 (Image from[**Kidney Recognition in CT using YOLOv3**](https://arxiv.org/pdf/1910.01268.pdf))

#### **Agriculture**

Robotics and artificial intelligence are very important in modern agriculture. Vision-based harvesting robots were developed to take the role of people manually picking fruits and vegetables. One of the top models in this industry makes use of YOLO. The authors of Tomato detection based on modified YOLOv3 framework explain how they used YOLO to determine which fruits and vegetables were best suited for harvesting.



Image from Tomato detection based on modified YOLOv3 framework ([**source**](https://www.nature.com/articles/s41598-021-81216-5))

### **3 - Security surveillance**

### Although security monitoring is the primary use for object detection, it is not the only one. YOLOv3 has been used to quantify social distance violations between people during the COVID19 epidemic.

### A deep-learning-based social distance monitoring approach for COVID-19 offers more information on this subject.

### **4 - Self-driving cars**

Real-time object detection is ingrained in autonomous vehicle systems from the beginning. Because autonomous vehicles must accurately recognise the correct lanes, all nearby objects, and pedestrians to maximise road safety, this integration is essential. When compared to straightforward image segmentation methods, YOLO is a better contender due to its real-time feature.