# **A Gentle Introduction to Object Recognition With Deep Learning**

* **Object recognition** is refers to a collection of related tasks for identifying objects in digital photographs.
* **Region-Based Convolutional Neural Networks, or R-CNNs**, are a family of techniques for addressing object localization and recognition tasks, designed for model performance.
* **You Only Look Once, or YOLO**, is a second family of techniques for object recognition designed for speed and real-time use.

**What is Object Recognition?**

Object recognition is a general term to describe a collection of related computer vision tasks that involve identifying objects in digital photographs.

***Image classification*** involves predicting the class of one object in an image. ***Object localization*** refers to identifying the location of one or more objects in an image and drawing abounding box around their extent. ***Object detection*** combines these two tasks and localizes and classifies one or more objects in an image.

As such, we can distinguish between these three computer vision tasks:

* **Image Classification**: Predict the type or class of an object in an image.

*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).

* **Object Localization**: Locate the presence of objects in an image and indicate their location with a bounding box.

*Input*: An image with one or more objects, such as a photograph.

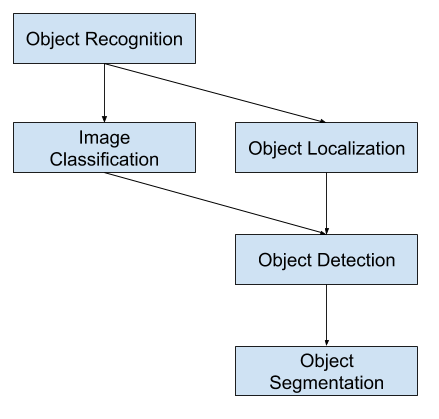
*Output*: One or more bounding boxes (e.g. defined by a point, width, and height).

* **Object Detection**: Locate the presence of objects with a bounding box and types or classes of the located objects in an image.

*Input*: An image with one or more objects, such as a photograph.

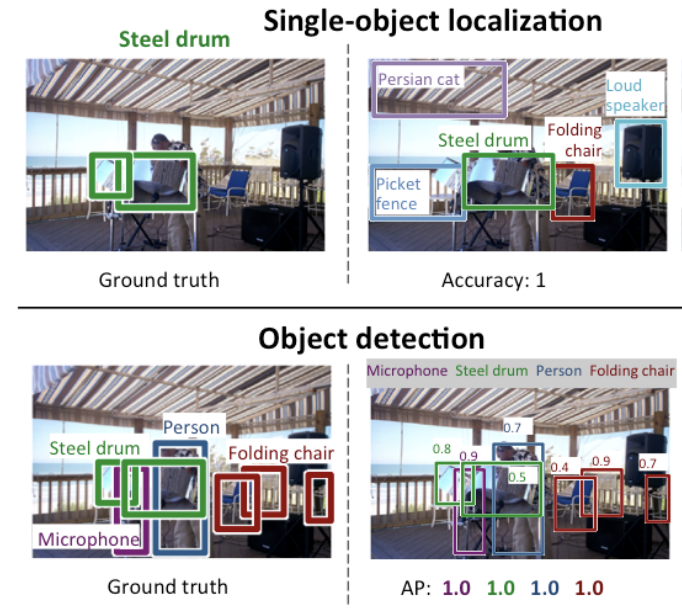
*Output*: One or more bounding boxes (e.g. defined by a point, width, and height), and a class label for each bounding box.

From this breakdown, we can see that object recognition refers to a suite of challenging computer vision tasks.



* **Image classification**: Algorithms produce a list of object categories present in the image.
* **Single-object localization**: Algorithms produce a list of object categories present in the image, along with an axis-aligned bounding box indicating the position and scale of one instance of each object category.
* **Object detection**: Algorithms produce a list of object categories present in the image along with an axis-aligned bounding box indicating the position and scale of every instance of each object category.

Below is an example comparing **single object localization** and **object detection**. Note the difference in ground truth expectations in each case.



The performance of a model for image classification is evaluated using the mean classification error across the predicted class labels. The performance of a model for single-object localization is evaluated using the distance between the expected and predicted bounding box for the expected class. Whereas the performance of a model for object recognition is evaluated using the precision and recall across each of the best matching bounding boxes for the known objects in the image.

## R-CNN Model Family

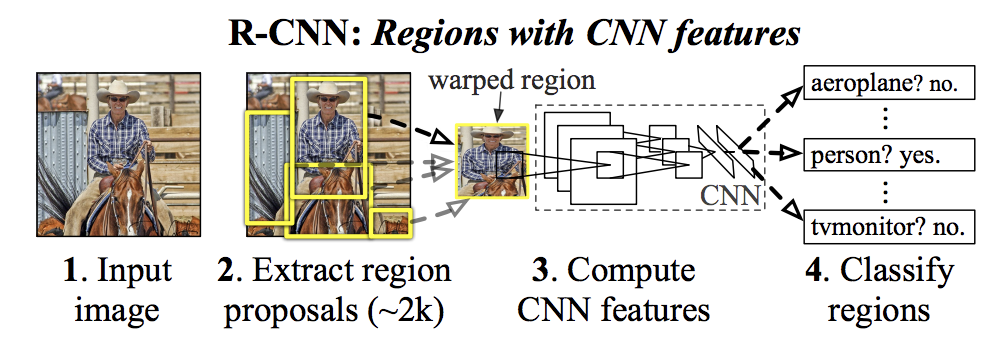
This includes the techniques R-CNN, Fast R-CNN, and Faster-RCNN designed and demonstrated for **object localization** and **object recognition**.  
Let’s take a closer look at the highlights of each of these techniques in turn.

### R-CNN

Their proposed R-CNN model is comprised of three modules; they are:

* **Module 1: Region Proposal**. Generate and extract category independent region proposals, e.g. candidate bounding boxes.
* **Module 2: Feature Extractor**. Extract feature from each candidate region, e.g. using a deep convolutional neural network.
* **Module 3: Classifier**. Classify features as one of the known class, e.g. linear SVM classifier model.

The architecture of the model is summarized in the image below.



A downside of the approach is that **it is slow**, requiring a CNN-based feature extraction pass on each of the candidate regions generated by the region proposal algorithm.

### Fast R-CNN

A review of the limitations of R-CNN, which can be summarized as follows:

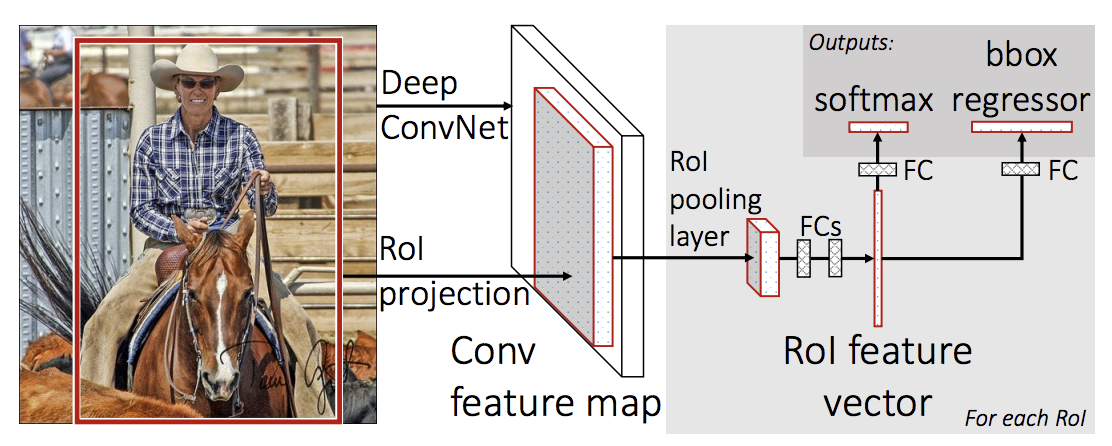
* **Training is a multi-stage pipeline**. Involves the preparation and operation of three separate models.
* **Training is expensive in space and time**. Training a deep CNN on so many region proposals per image is very slow.
* **Object detection is slow**. Make predictions using a deep CNN on so many region proposals is very slow.

Fast R-CNN is proposed as a single model instead of a pipeline to learn and output regions and classifications directly. This did speed up the extraction of features, but essentially used a type of forward pass caching algorithm.

The architecture of the model takes the photograph a set of region proposals as input that are passed through a deep convolutional neural network. A pre-trained CNN is used for feature extraction. The end of the deep CNN is a custom layer called a Region of Interest Pooling Layer, or RoI Pooling, that extracts features specific for a given input candidate region.

The output of the CNN is then interpreted by a fully connected layer then the model bifurcates into two outputs, one for the class prediction via a softmax layer, and another with a linear output for the bounding box. This process is then repeated multiple times for each region of interest in a given image.

The architecture of the model is summarized in the image below.



The model is significantly faster to train and to make predictions, yet still requires a set of candidate regions to be proposed along with each input image.

### Faster R-CNN

The model architecture was further improved for both speed of **training** and **detection**.

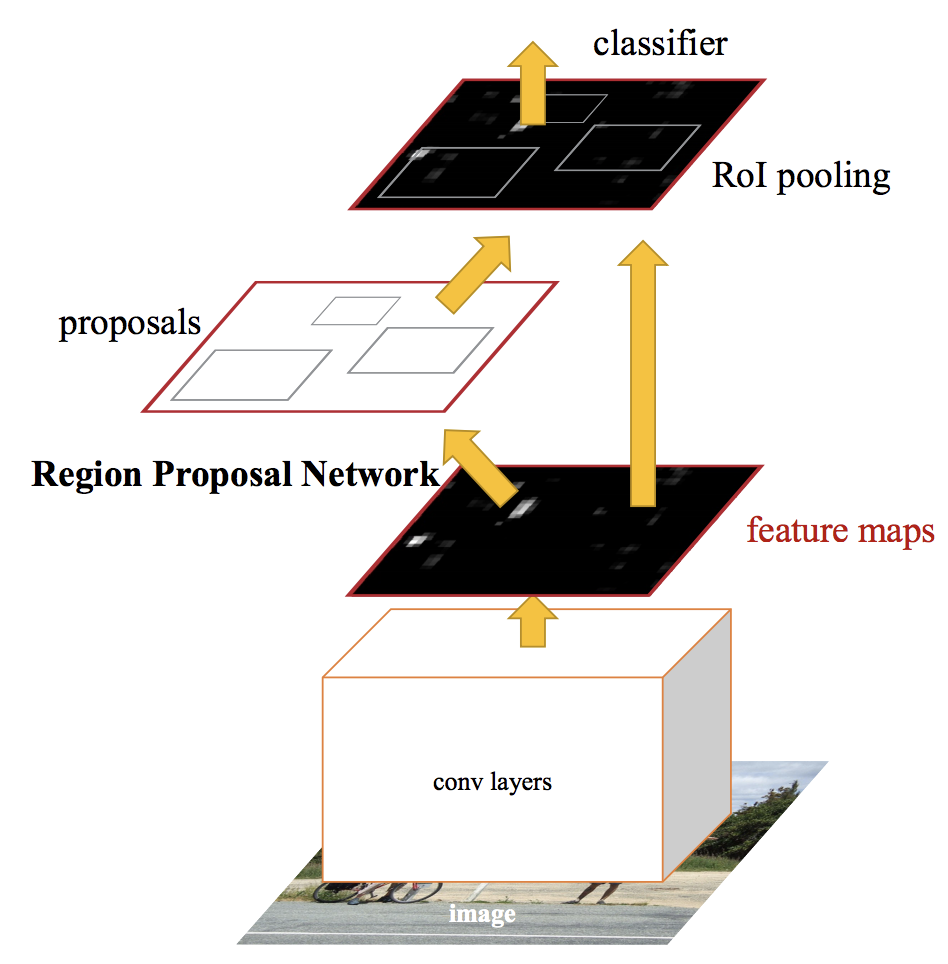
The architecture was designed to both propose and refine region proposals as part of the training process, referred to as a Region Proposal Network, or RPN. These regions are then used in concert with a Fast R-CNN model in a single model design. These improvements both reduce the number of region proposals and accelerate the test-time operation of the model to near real-time with then state-of-the-art performance.

Although it is a single unified model, the architecture is comprised of two modules:

* **Module 1: Region Proposal Network**. Convolutional neural network for proposing regions and the type of object to consider in the region.
* **Module 2: Fast R-CNN**. Convolutional neural network for extracting features from the proposed regions and outputting the bounding box and class labels.

Both modules operate on the same output of a deep CNN. The region proposal network acts as an attention mechanism for the Fast R-CNN network, informing the second network of where to look or pay attention.

The architecture of the model is summarized in the image below.



The RPN works by taking the output of a pre-trained deep CNN and passing a small network over the feature map and outputting multiple region proposals and a class prediction for each. Region proposals are bounding boxes, based on so-called anchor boxes or pre-defined shapes designed to accelerate and improve the proposal of regions. The class prediction is binary, indicating the presence of an object, or not, of the proposed region.

## YOLO Model Family

Another popular family of object recognition models is referred to collectively as YOLO or “You Only Look Once”.

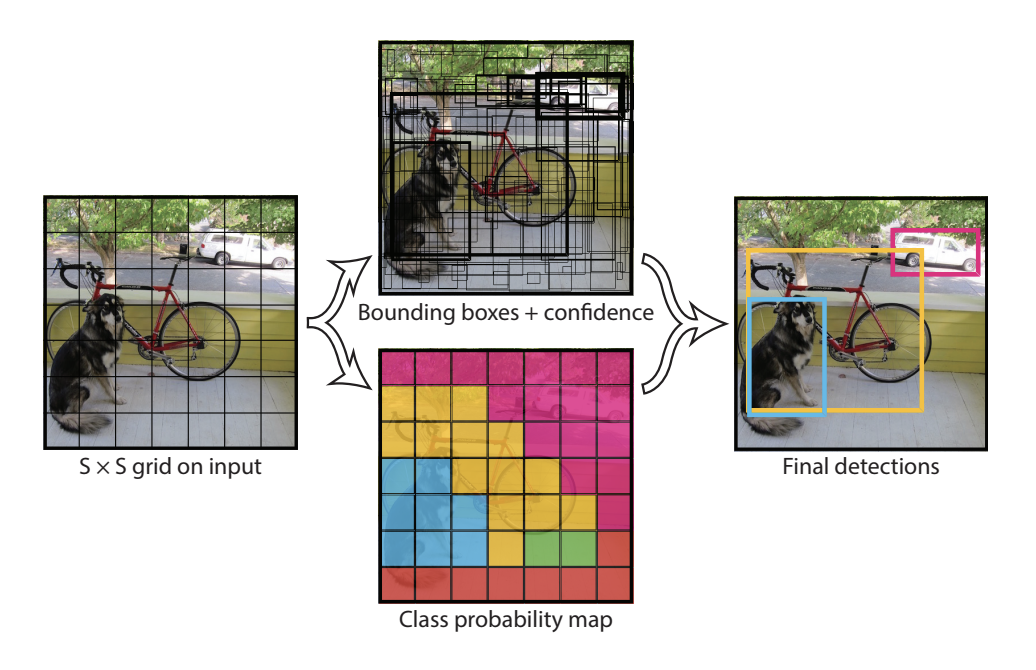
**The R-CNN models** may be generally **more accurate**, yet the **YOLO family** of models are fast, **much faster than R-CNN**, achieving object detection in real-time.

### YOLO

The approach involves a single neural network trained end to end that takes a photograph as input and predicts bounding boxes and class labels for each bounding box directly. The technique offers lower predictive accuracy (e.g. more localization errors).

The model works by first splitting the input image into a grid of cells, where each cell is responsible for predicting a bounding box if the center of a bounding box falls within it. Each grid cell predicts a bounding box involving the x, y coordinate and the width and height and the confidence. A class prediction is also based on each cell.

For example, an image may be divided into a 7×7 grid and each cell in the grid may predict 2 bounding boxes, resulting in 94 proposed bounding box predictions. The class probabilities map and the bounding boxes with confidences are then combined into a final set of bounding boxes and class labels. The image taken from the paper below summarizes the two outputs of the model.

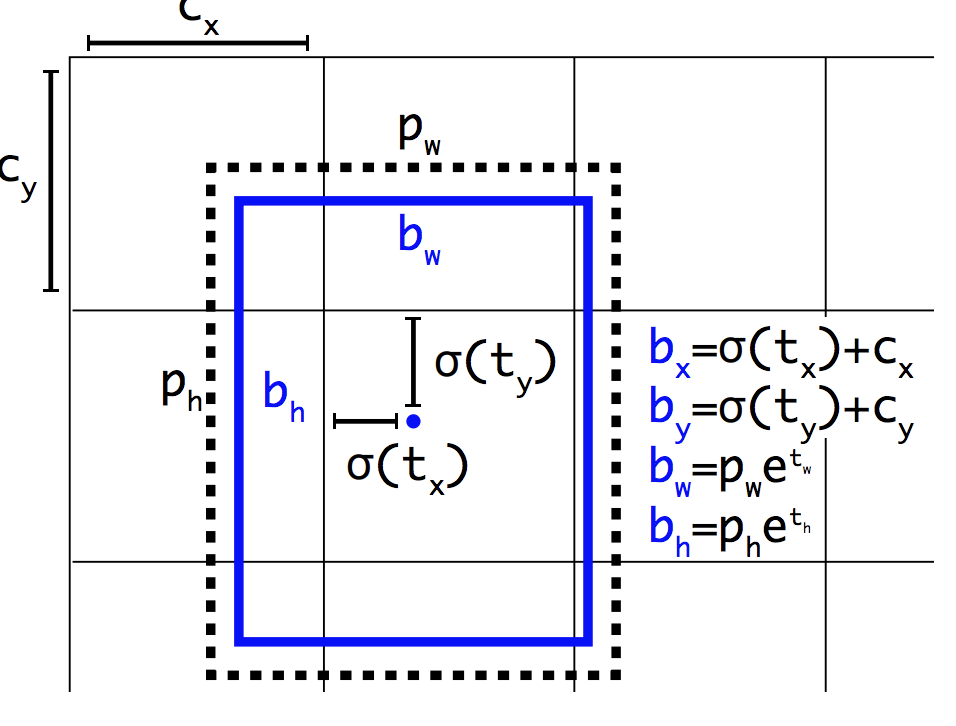


### YOLOv2 (YOLO9000) and YOLOv3

Although this variation of the model is referred to as YOLO v2, an instance of the model is described that was trained on two object recognition datasets in parallel, capable of predicting 9,000 object classes, hence given the name “YOLO9000.”

Like Faster R-CNN, YOLOv2 model makes use of anchor boxes, pre-defined bounding boxes with useful shapes and sizes that are tailored during training. The choice of bounding boxes for the image is pre-processed using a k-means analysis on the training dataset.

Importantly, the predicted representation of the bounding boxes is changed to allow small changes to have a less dramatic effect on the predictions, resulting in a more stable model. Rather than predicting position and size directly, offsets are predicted for moving and reshaping the pre-defined anchor boxes relative to a grid cell and dampened by a logistic function.



**Summary**

In this post, you discovered a gentle introduction to the problem of object recognition and state-of-the-art deep learning models designed to address it.

Specifically, you learned:

* **Object recognition** refers to a collection of related tasks for identifying objects in digital photographs.
* **Region-Based Convolutional Neural Networks, or R-CNNs**, are a family of techniques for addressing object localization and recognition tasks, designed for model performance.
* **You Only Look Once, or YOLO**, is a second family of techniques for object recognition designed for speed and real-time use.