**YOLO**

Introduction:

Yolo (“You Only look Once”) is an effective real-time object recognition algorithm, first described in the seminal [2015 paper by Joseph Redmon et al.](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf) This algorithm uses neural networks concept for real-time object detection in images and videos.

Key features of YOLO are as follows:

* Yolo is solved as a regression problem and provides class probabilities of the detected images.
* The algorithm requires only a single forward propagation through neural network to detect objects, thus improving the **speed** of object detection process.
* Yolo is **highly accurate** with minimal background errors.
* Yolo has very good **learning capability** which enables to learn the representation of objects and apply them in object detection.

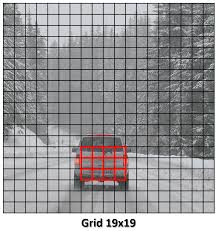
Working:

Yolo algorithm is a culmination of three techniques.

* Residual Blocks.
* Bounding box regression.
* Intersection Over Union (IOU).

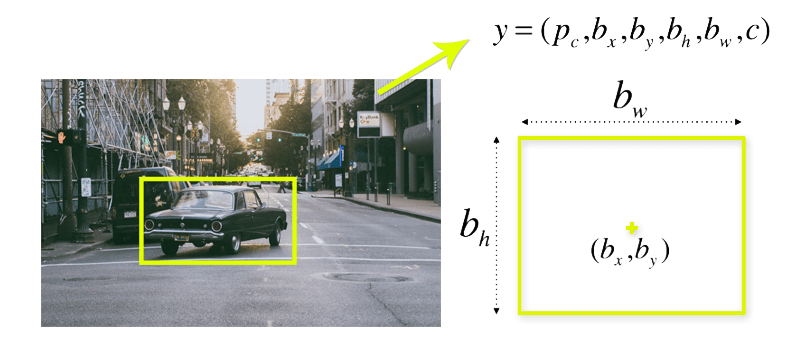
Residual Blocks :

* The given image is divided in various grids each with dimension SxS.
* All grid are of equal sizes.
* Every grid detects objects which appear in them.
* The grid having center of an object is responsible for detecting the object.



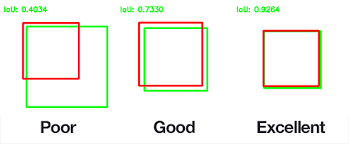
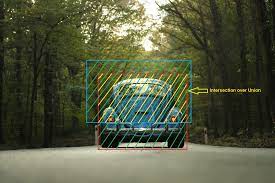
Bounding Box Regression:

* This is used to highlight the exact location of the object in a given image.
* Attributes related to the bounding boxes are as follows:
  + Width(bw).
  + Height(bh).
  + Class – (An identified object ).
  + Bounding box centre (bx,by).
* Yolo uses a single bounding bound regression to predict all these attributes with respect to the object.



Intersection Over Union:

* This technique is used to describe how different boxes overlap during object detection.
* YOLO uses IOU to provide an output box that surrounds the objects perfectly.
* Each grid cell will predict the bounding box for the object with certain confidence level.
* If the predicted box is similar to the real box then the output of IOU is 1.
* This helps us in eliminating all the bounding boxes which are not equal to the real one.



Versions of YOLO:

Yolo mainly has five versions, namely YOLO V1, YOLO V2, YOLO V3, YOLO V4, YOLO V5.

The difference between these versions with respect to improvement measures taken from V1-V5 are as follows:

* YOLO - The grid division is responsible for detection, confidence loss.
* YOLO V2 - Anchor with K-means added, two-stage training, full convolutional network.
* YOLO V3 - Multi-scale detection by using FPN.
* YOLO V4 - SPP, MISH activation function, data enhancement Mosaic/Mixup, GIOU(Generalized Intersection over Union) loss function.
* YOLO V5 - Flexible control of model size, application of Hardswish activation function, and data enhancement.

YOLO V4:

I have used YOLO V4 in this assignment. The [fourth version](https://github.com/AlexeyAB/darknet) of the YOLO algorithm was released in April 2020 by Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao in their article [“YOLOv4: Optimal Speed and Accuracy of Object Detection”](https://arxiv.org/pdf/2004.10934.pdf).

YOLO V4 style has significant changes when compared to YOLO V3. More focus on comparing data, and has a substantial improvement. The integrator characterizes it and fully achieves very high performance. We can summarize it like this:

YOLO V4=CSP Darknet53+SPP+Pan+YOLO V3.

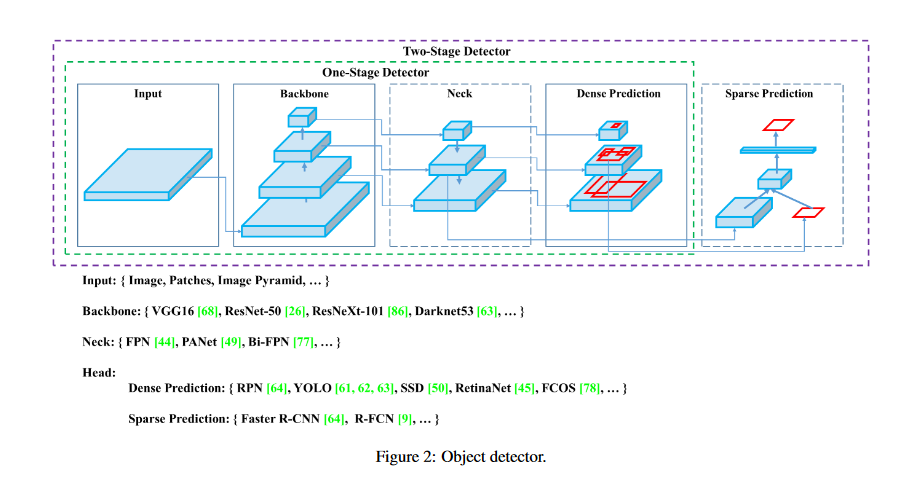
Some other characteristics of YOLO V4 are as follows:

* Here, several anchor points are responsible for one ground truth unlike in YOLO V3 where one anchor point is responsible to for one ground truth.
* The CIOU(Complete Intersection Over Union) loss function is incorporated.

Architecture of YOLO V4:

The architecture consists of various parts, broadly they are - The input which comes first and it is basically what we've as our set of training images which will be fed to the network - they are processed in batches in parallel by the GPU. Next are the Backbone and the Neck which do the feature extraction and aggregation. The Detection Neck and Detection Head together can be called as the Object Detector.

And finally, the head does the detection/prediction. Mainly, the Head is responsible for the detection (both localization and classification).



Backbone Network :

Uses CSPDarkNet53 which is based on DenseNet design. It has **Dense connectivity pattern**, which helps it to concatenate the previous inputs with the current input before proceeding into the dense layers.

CSPDarkNet53 has two blocks namely Convolutional Base Layer and Cross Stage Partial (CSP) Block.

Neck:

1. SPP – Additional Block:
   * An additional block called **SPP (Spatial Pyramid Pooling)** is added in between the CSPDarkNet53 backbone and the feature aggregator network (PANet), this is done to increase the receptive field and separates out the most significant context features and has almost no effect on network operation speed.
2. PANet (Path Aggregation Network):
   * YOLOv4 uses a modified path aggregation network, mainly as a design improvement in-order to make it more suitable for training on a single GPU.
   * The main role of PANet is to improve the process of instance segmentation by keeping the spatial information which in turn helps in proper localization of pixels for mask prediction.

Head:

* The main function here is locating bounding boxes and performing classification.
* It has a similar functionality as that of YOLO V3.

Conclusion:

* We discussed basics of YOLO.
* Different Versions in YOLO.
* YOLO V4 and it’s architecture.

References:

* <https://pjreddie.com/darknet/yolo/>
* <https://reader.elsevier.com/reader/sd/pii/S1877050922001363?token=9AF8C6B9DF91B3036A447969A960FEE285278C3F29BD66C202305B8267C4ABF94FBF73CBF265148E00E795802DE1F880&originRegion=eu-west-1&originCreation=20220902161143>
* <https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/>
* <https://appsilon.com/object-detection-yolo-algorithm/>
* <https://arxiv.org/abs/1506.02640>
* <https://towardsdatascience.com/whats-new-in-yolov4-323364bb3ad3>
* <https://iq.opengenus.org/yolov4-model-architecture/>
* <https://pyimagesearch.com/2022/04/11/understanding-a-real-time-object-detection-network-you-only-look-once-yolov1/#:~:text=Network%20Architecture%20of%20YOLO,-The%20network%20architecture&text=It%20consists%20of%20mainly%20three,box%20coordinates%20and%20classification%20scores>.
* <https://arxiv.org/pdf/2004.10934.pdf>