YOLOv4 Architecture

YOLOv4 is known for its upgradation in terms of AP and FPS. YOLOv4 prioritizes real-time object detection and training takes place on single CPU. YOLOv4 has obtained state-of-art results on COCO dataset with 43.5% speed (AP) at 65 Performance (FPS) on Tesla V100. This achievement is the result of combination of the features like DropBlock Regularization, Data Augmentation, Mish-Activation, Cross-Stage-Partial-connections (CSP), Self-adversarial-training (SAT), Weighted-Residual-Connections (WRC) and many more.

There are two types of models, one and two staged object detectors. In two stage detectors works in two parts that is first regions of importance are detected and then regions are classified to see if object is detected in that particular region. YOLOv4 being single staged object detector, works more accurate and faster than Two staged detectors like R-CNN, Fast R-CNN.

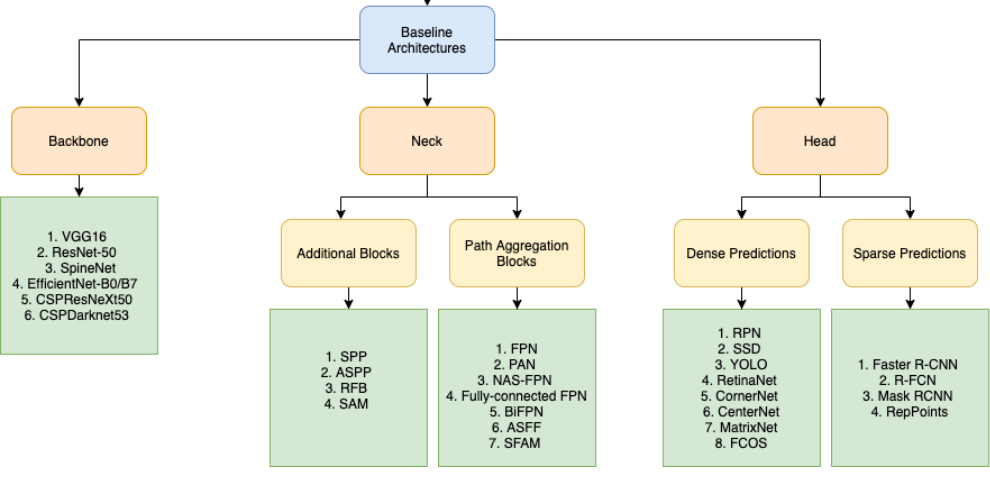


Fig 2.2. Basic YOLOv4 Architecture

*Backbone:*

YOLOv4 basically uses one of the three model as its backbone. Three feature extractor models include:

CSPResNext50:

Both CSPResNext50 and CSPDarknet53 are DenseNet based models. It works similar to CSPDarknet53 that operates on CSPNet strategy. Considering COCO dataset, CSPDarknet53 is better in classifying objects than CSPResNext50. CSPResNext50 consists of 16 CNN layers with 425×425 receptive field and 20.6 M parameters while CSPDarknet53 consists of 29 CNN layers with 725×725 receptive field and 27.6 M parameters.

CSPDarknet53:

It is widely used Backbone of Object detection that makes use of DarkNet-53. YOLOv4 Specifically uses CSPDarknet53 as its backbone. It operates on a CSPNet strategy of dividing DenseBlock consisting feature map in two halves and then merging them together via cross-stage hierarchy. The former part circumvents the base layer and is used as input of the next transition layer. The later part of base layer undergoes DenseBlock. This strategy decreases the computational complexity. It has accuracy that is better when compared to other ResNet models and hence has good performance.

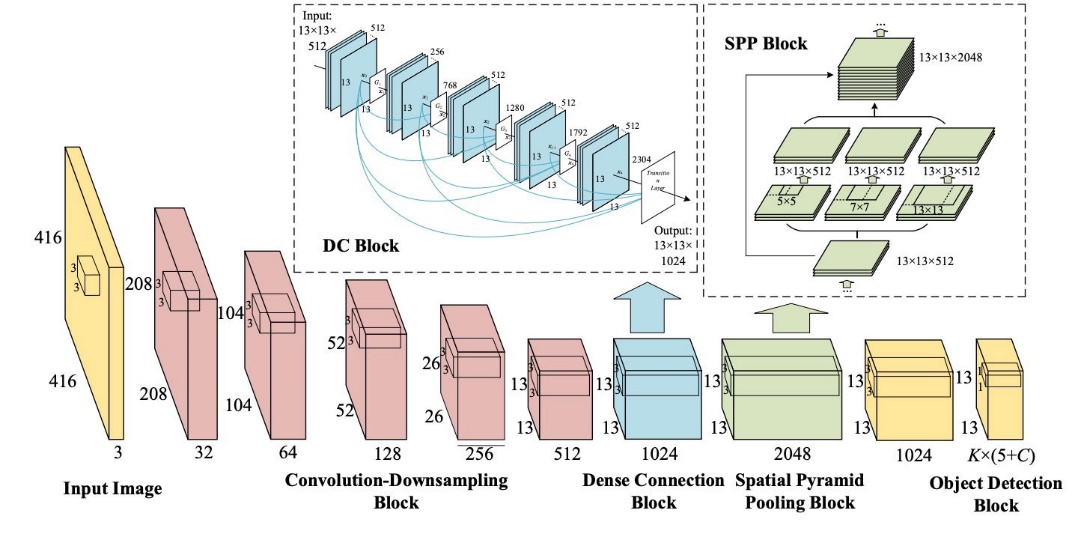


Fig- CSPDarknet53

EfficientNet-B3:

It is used as image Classification Model that is particularly used to attain state-of-art accuracy. It is generally utilized to restudy the Convolutional Neural Network scaling and is based on AutoML. AutoML mobile Framework was developed in order to develop a small sized network known as EfficientNet-B0. The Compounding Scaling as name suggests helps in scaling up AutoML baseline in order to gain Efficient-B1 to Efficient-B7

*Neck:*

This is the second stage where features are gathered that where initially formed ­­in the backbone and later these features will be fed to head for detection. YOLO­v4 has several options:

FPN (Feature Pyramid Networks)

Earlier, detectors used to predict objects on pyramidal feature hierarchy extracted from backbone. In order to solve this major issue of effective representation and multi-scale feature processing, FPN was proposed that follows top down path to gather features of different scales.

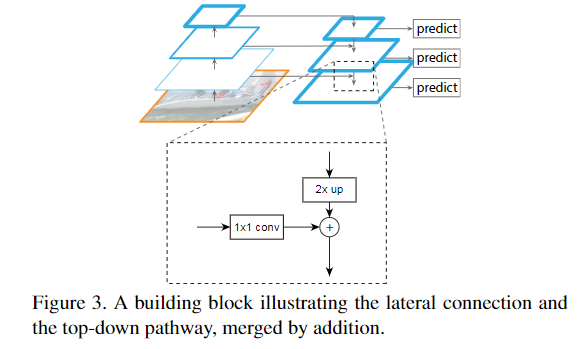


Fig- FPN (Feature Pyramid networks)

PAN (Path Aggregation Network)

It is used as neck in YOLOv4 Algorithm and is used to improve the process of segmentation by maintaining semantic data or information precisely that helps in correct localization of picture elements for mask information.

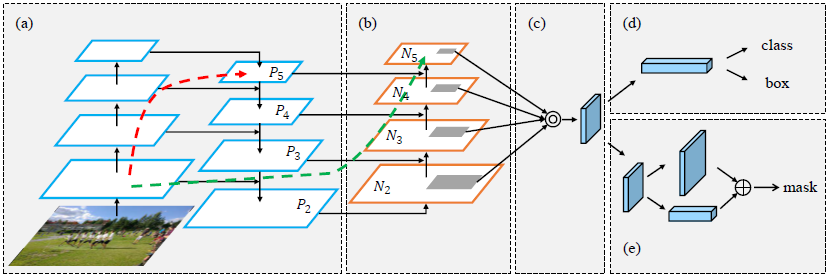


Fig- PANet Architecture

Properties of PAN:

Bottom-up Path Augmentation:

When image is passed through various layers of Neural network, spatial resolution of image reduces while complexity of the feature increases simultaneously. This results in complications in identification of pixel-level masks by high level features. In YOLOv3 FPN follows top down approach for combining multi-scale features preserving semantic localization information. Huge objects mask production becomes intricate and lengthy because the information needs to travel a greater number of layers.

On contrary, PANet includes an additional bottom up and top down route than FPN. This considerably reduces the complexity by usage of lateral connections from lower layers to upper. It has at most 10 years which makes it less complicate and lengthy.

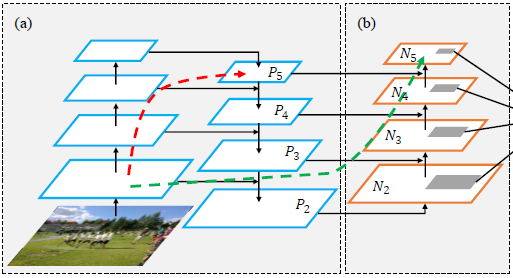


Fig- Bottom-up Path Augmentation

Adaptive Feature Pooling:

Earlier, techniques like Mask-RCNN made used of features from only one stage for prediction. RIO Align Pooling was used to uproot features from upper levels. At times result were predicted inaccurately. So, in order to overcome this problem PANet extracted features from all the layers which performs Align Pooling on every feature map to uproot features.

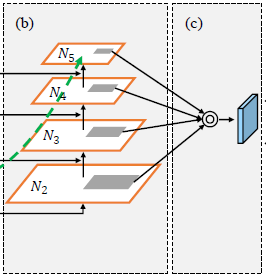


Fig- Adaptive Feature Pooling

Fully-Connected Fusion:

Fully Convolutional Network was used in Mask-RCNN to conserve spatial information and simultaneously number of parameters are also reduced. The model cannot learn how to use pixel location for prediction since the parameters are shared for all the spatial positions.Fully-connected layers are location sensitive and can adjust with different locations.PANet uses data from both layers for precise prediction.

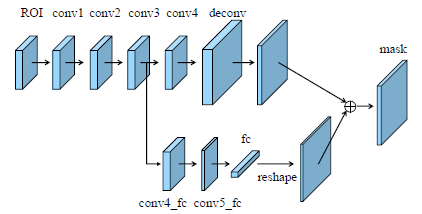


Fig- Fully-Connected Fusion

SPP (Spatial Pyramid Pooling)

SPP also known as Spatial Pyramid Pooling is used for obtaining both fine and coarse information. Sliding Kernel of dimension 1×1, 5×5, 9×9, 13×13 to which pool is applied. Maps of various kernel sizes are combined to get output. The main advantage to using SPP is to improve receptive field. Spatial Pyramid Pooling is used to create fixed size features regardless of feature map sizes.

Advantages of SPP are it creates fixed result whatever might be input feature map size. Also, it makes use of multi-level spatial bins hence it is robust when it comes to object deformations. SPP is flexible in terms of input scales as it can extract pool features at variable scales.

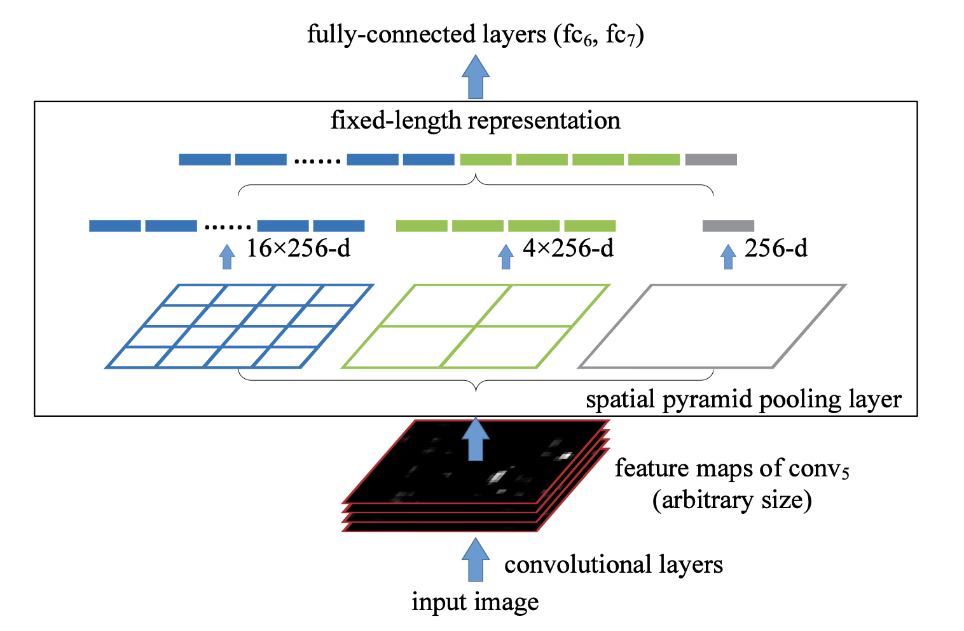


Fig- Spatial Pyramid Pooling

*Head:*

The main objective of the head in YOLOv4 is to perform prediction that includes classification and regression of bounding boxes. It uses YOLOv3 head. It provides information regarding coordinates of bounding boxes (x, y, h, w). It includes width, height, center and score of prediction with label. YOLOv4 head can be applied to every anchor box.

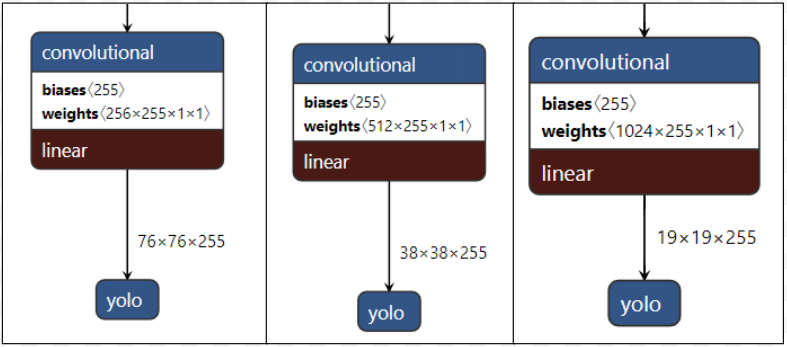


Fig YOLOv4 head applied at different scales

*Anchor box:*

Anchor boxes are used for multiple objects of various sizes in a single frame whose center is positioned in the same cell. On contrary, grid was used to detect single object in a frame. If number of anchor boxes change then length of ground truth and prediction array also changes.

Consider a box in cell has 80 prediction classes i.e. [Pc, P1, P2... P80, X1, Y1, X2, Y2] which totals to 85, for these 9 anchor boxes will be used and will have array length of 85\*9 = 765 predictions.

Example of anchor box plated around (0,0) of different scales.

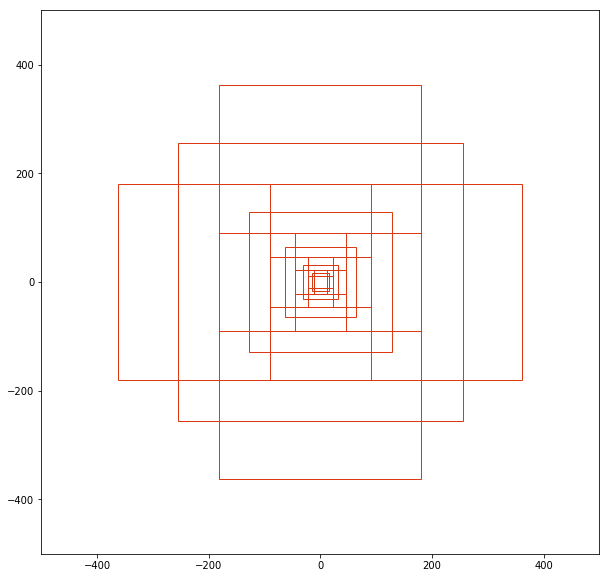


Fig Anchor boxes

*Bag of Freebies YOLOv4*

Bag of Freebies can be termed as techniques that improvise cost related to training or strategy to ameliorate the accuracy of model. It increases the performance of model without compromising the latency at inference time and hence improvements are seen in data management and data augmentation. These methods enhance and augment the training dataset of model that are opened to situations that otherwise might have unsensed. Data Augmentation hence can be seen as a strategy to get utmost information from the dataset and would overall generalize the data or increase the robustness so that model can learn through a shallow range of situations.

*Bag of Specials YOLOv4*

Bag of specials consists of coherence and post-processing models due to which inference cost of the modules escalate by a fraction. However, this change helps the detector improve the accuracy and performance. Selection of the technique totally depends on architecture and other technical parameters but end outcome of refining is attained.

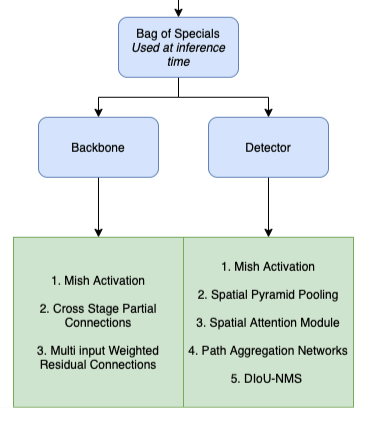


Fig- Different methods present in Bag of Specials.