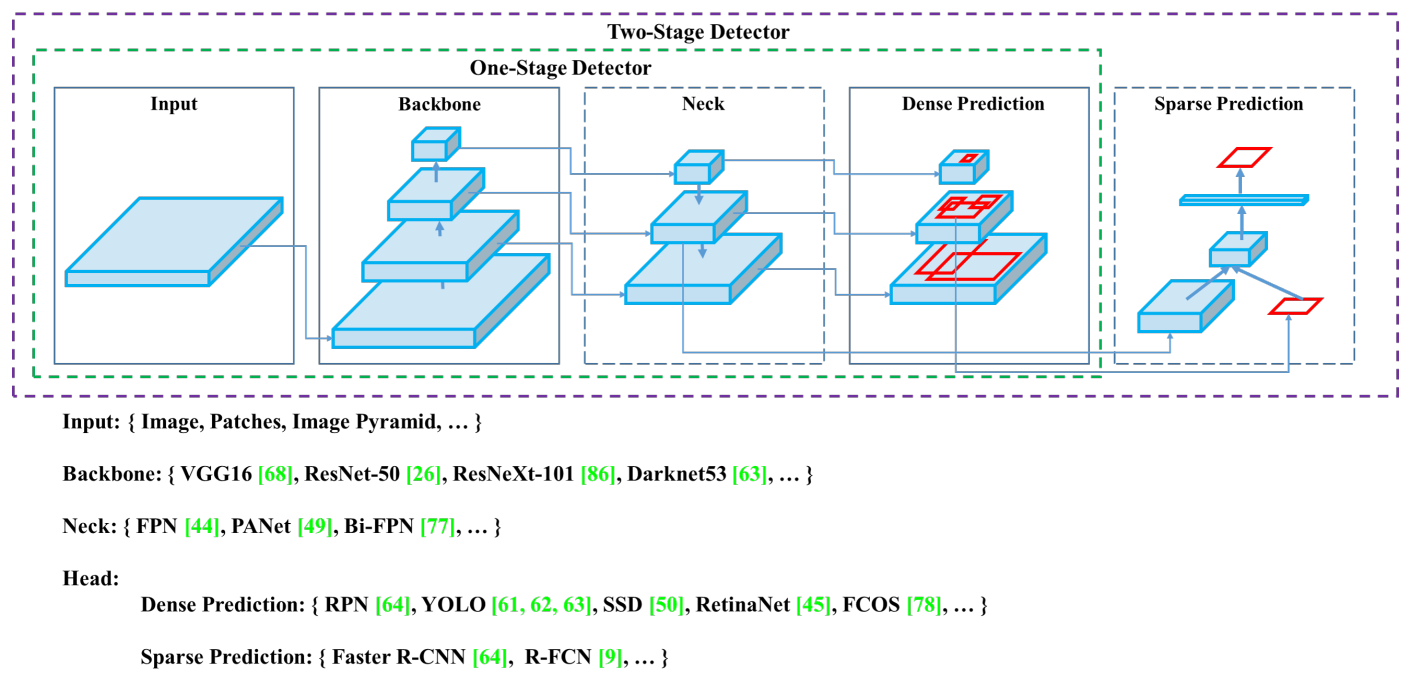
**YOLOv4: Optimal Speed and Accuracy of Object Detection**

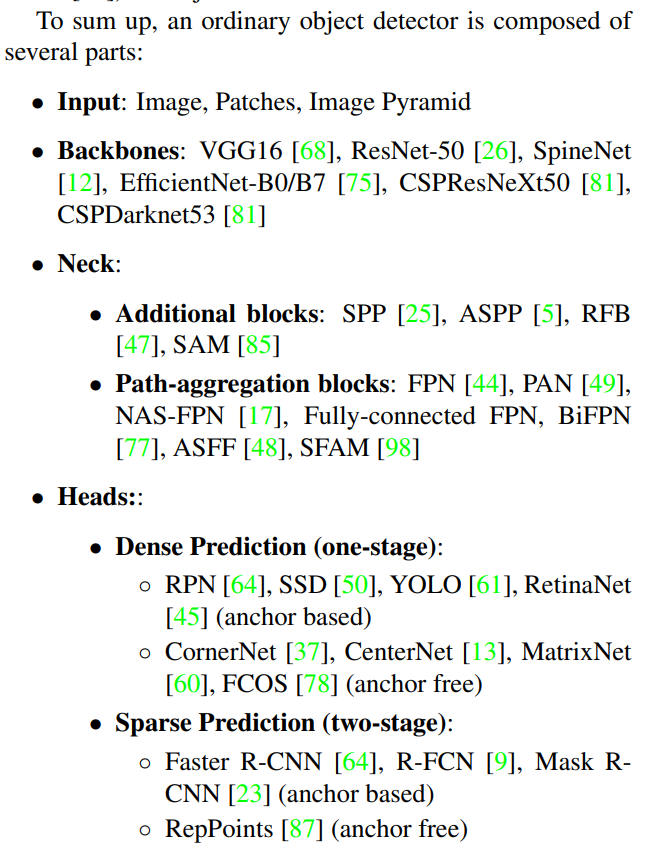
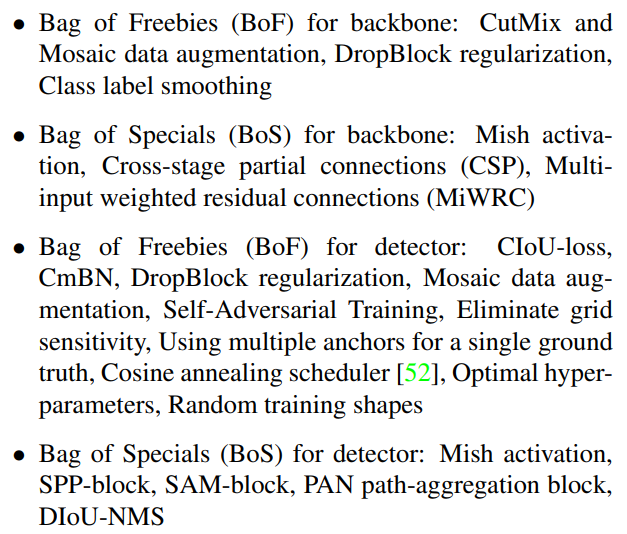
*Alexey Bochkovskiy Chien-Yao Wang Hong-Yuan Mark Liao*

* Source: <https://arxiv.org/pdf/2004.10934.pdf>
* Considers multiple universal features:

Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross mini-Batch Normalization (CmBN), Self-adversarial-training (SAT), Mish-activation, Mosaic data augmentation, DropBlock regularization, CIoU loss, etc.

* *The most accurate modern neural networks do not operate in real time and require large number of GPUs for training with a large mini-batch-size. We address such problems through creating a CNN that operates in real-time on a conventional GPU, and for which training requires only one conventional GPU.*
* Contributions:
  + *We develop an efficient and powerful object detection model. It makes everyone can use a 1080 Ti or 2080 Ti GPU to train a super-fast and accurate object detector.*
  + *We verify the influence of state-of-the-art Bag-of-Freebies and Bag-of-Specials methods of object detection during the detector training.*
  + *We modify state-of-the-art methods and make them more efficient and suitable for single GPU training, including CBN [89], PAN [49], SAM [85], etc.*



* *A modern detector is usually composed of two parts, a backbone which is pre-trained on ImageNet and a head which is used to predict classes and bounding boxes of objects.*
* If using GPU, the backbone could be VGG, ResNet, ResNeXt, DenseNet, etc. If using CPU, the backbone could be SqueezeNet, MobileNet, ShuffleNet, etc.
* Head could be one-stage or two-stage detector. Two-state detectors include RCNN, Fast RCNN, Faster RCNN, Libra RCNN, RepPoints (anchor-free), etc. One-stage detectors include YOLO, SSD, RetinaNet, etc. Anchor free one-stage detectors include CenterNet, CornerNet, FCOS, etc.
* Often there are a few layers between backbone and head, and they collect feature maps from different stages. These are together called as neck.
* A neck has bottom-up and top-down paths. E.g. Feature Pyramid Network (FPN), Path Aggregation Network (PAN), BiFPN, etc.
* 
* Bag of Freebies:
  + Techniques used during training to improve the model accuracy without affecting the model computation cost during inference.
  + One example is data augmentation.
  + Photometric distortions: changing brightness, contrast, hue, saturation, noise, etc.
  + Geometric distortions: cropping, scaling, rotating, flipping, etc.
  + Above are pixel-wise data augmentation techniques.
  + Simulating object occlusion: random erase, CutOut, hide-and-seek, grid mask
  + Occluding feature maps: DropOut, DropConnect, DropBlock
  + Combining multiple images: MixUp, CutMix
  + Handling data imbalance:
    - Hard negative example mining and online hard example mining for two-stage detectors
    - Focal loss for one-stage detector.
  + Label smoothing
  + For bounding box regressors: GIoU, DIoU, CIoU losses
* Bag of Specials:
  + Plugin modules and post-processing methods that only increase the inference cost by a small amount but can significantly improve the accuracy of object detection.
  + These modules generally enlarge receptive field, introduce attention mechanism, strengthen feature integration, etc.
  + Postprocessing is also considered bag-of-specials technique.
  + Enhance receptive field: SPP, ASPP, RFB.
  + Attention module: channel-wise and point-wise attention mechanism; e.g. Squeeze-and-Excitation (SE), Special Attention Module (SAM)
  + Feature integration: Skip connection, hyper-column, FPN
  + Activation function: ReLU, LReLU, PReLU, ReLU6, Scaled Exponential Linear Unit (SELU), Swish, hard-Swish, Mish
  + NMS
* CSPResNext50 is considerably better compared to CSPDarknet53 in terms of object classification on the ILSVRC2012 (ImageNet) dataset. However, CSPDarknet53 is better compared to CSPResNext50 in terms of detecting objects on the MS COCO dataset.
* *A reference model which is optimal for classification is not always optimal for a detector.*
* *A model with a larger receptive field size (with a larger number of convolutional layers 3 × 3) and a larger number of parameters should be selected as the backbone.*
* YOLOv4:
  + *Backbone: CSPDarknet53*
  + *SPP block over the CSPDarknet53 for increasing the receptive field, separating out the most significant context features, and causing almost no reduction of the network operation speed.*
  + *PANet as for parameter aggregation from different backbone levels for different detector levels, instead of the FPN used in YOLOv3.*
  + *Head: YOLOv3 (anchor based)*
* *PReLU and SELU are more difficult to train, and ReLU6 is specifically designed for quantization network.* So, they are not used.
* DropBlock for regularization
* syncBN is not considered for normalization since we want single GPU training
* Mosaic and Self-Adversarial Training (SAT) data augmentation
* Optical hyper-parameters using genetic algorithms
* Modified SAM, PAN, CmBN
* Mosaic data augmentation: mixes 4 training images. Allows detections of objects outside their normal context.
* *Self-Adversarial Training (SAT) also represents a new data augmentation technique that operates in 2 forward backward stages. In the 1st stage the neural network alters the original image instead of the network weights. In this way the neural network executes an adversarial attack on itself, altering the original image to create the deception that there is no desired object on the image. In the 2nd stage, the neural network is trained to detect an object on this modified image in the normal way.*
* **
* TODO: Page 8 onwards