# YOLO v4 Highlights ([Article](https://jonathan-hui.medium.com/yolov4-c9901eaa8e61) on Medium)

43.5% AP at 65 FPS for the Microsoft COCO test-dataset.

## BAG OF FREEBIES (IMPROVED TRAINING STRATEGIES FOR THE BACKBONE NETWORK):

All are regularization techniques for training-phase.

1. CutMix Data Augmentation:
   1. Removes a part of the image and replaces it with dummy pixels or subset of another image.
   2. Helps avoid overfitting (Regularization).

Graphical user interface, application

Description automatically generated with medium confidence

1. Mosaic Data Augmentation:
   1. Combines 4 images into a single training-image to avoid overfitting (Regularization).

Timeline

Description automatically generated with medium confidence

1. DropBlock regularization:
   1. Analogous to Dropout in FC Neural nets.
   2. We drop an entire block of pixels from the train-image to avoid overfitting.

A picture containing crossword puzzle, dog

Description automatically generated

1. Class Label Smoothing:
   1. Avoid overfitting by reducing the threshold of class label during Loss computation.
   2. Used when 100 train images, model predicts all examples with 0.9 confidence, but accuracy is only 0.6.
   3. If K=3, target vector=[1,0,0], our model would try to predict something like [*10, 0, 0*] Using hyperparam the smoothed target vector is now=[0.933, 0.03, 0.03], and our model will predict [3.3, 0, 0].

## BACKBONE:

1. Dense Block:
   1. Each layer
   2. Each layer has input ouput of all previous layers

Diagram, engineering drawing

Description automatically generated

1. Dense Network:
   1. Multiple Dense Blocks interconnected by Transition Layers
   2. Each Transition Layer

A picture containing text, screenshot, device, gauge

Description automatically generated

1. Cross Stage Partial Connections (CSP):
   1. Instead of conventional Residual/Skip connections (RESNET-50 used in older YOLO v2), we divide the input feature map of each Dense Block into : Directly moves into the next Transition Layer , and Fed into each Dense Block.
   2. This ensures better gradient flow during backprop through different network flows.
   3. Also computational effort at bottlenecks reduced by 20%.

A picture containing graphical user interface

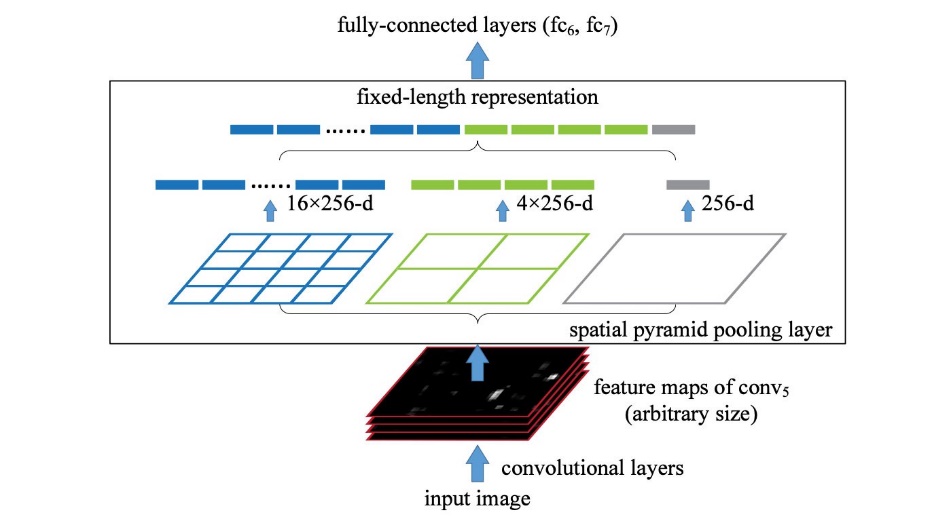
Description automatically generated

1. CSPDarknet53:
   1. CSP connections with Darknet-53 has higher accuracy in object detection compared with ResNet based designs, and with a better classification performance! Classification accuracy of CSPDarknet53 can be improved with *Mish*-like techniques below.
   2. YOLOv4 uses CSPDarknet53.
2. Neck:
   1. To detect objects at different scales, the head (final detection layer rightmost image) probes the feature-maps at different layers i.e. spatial resolutions.
   2. Neighboring feature maps from the bottom-up stream, and neighboring feature maps from the top-down stream are added together element-wise or concatenated before feeding into the head.
   3. Hence, the head’s input will contain **spatial rich information from the bottom-up stream** and the **semantic rich information from the top-down stream**. This part of the system is called a neck.

Diagram

Description automatically generated

1. SPP (Spatial Pyramid Pooling Layer): [Youtube](https://www.youtube.com/watch?v=2IoHC_fhrFU)
   1. YOLOv3 already implemented FPN (Feature Pyramid Networks) for detecting objects at multiple scales.
   2. Last layer is just replaced with SPN layer to improve accuracy of any model.
   3. SPP will avoid warping the input feature map into a different spatial dimension, and will apply max-pooling at Levels- 1x1, 2x2, and 4x4 to get 1-dimensional vectors of same size, as shown in the below image.
   4. It then concatenates all these 1-D vectors and pushes them to the usual Fully-Connected Layers for final prediction.



1. YOLOv4 uses SPP Blocks and Dense Blocks, both:

Diagram

Description automatically generated

## BAG OF SPECIALS (INCREASES INFERENCE QUALITY AT VERY SMALL COST):

1. Mish Activation:
   1. Combination of softplus and swish activation functions:

Chart, line chart

Description automatically generated

1. Multi-input Weighted Residual Connections (MiWRC):
   1. As shown below on left, the conventional Residual Block first extracts low-dimensional “core information” from input.
   2. However, in YOLOv4 we have inverse Residual Blocks where we expand input into higher-dimensions and then perform extraction of “core information”. We don’t perform non-linear transformations at the lower-dimensional input (to avoid loss of representation).

Diagram

Description automatically generated

1. CIoU Loss:
   1. If predicted bounding-boxes are non-overlapping with expected bounding-box, conventional IoU doesn’t work.
   2. Distance IoU is better because it takes into account the smallest box that covers these non-overlapping boxes (the predicted and expected).

\_\_\_where measures the Euclidean distance between the center of the 2 bounding-boxes and is the diagonal length of the smallest box covering both these boxes.

* 1. Finally, Combined IoU also takes into account the consistency of aspect ratios.
  2. DIoU is additionally used during the time of Non Max Suppression.