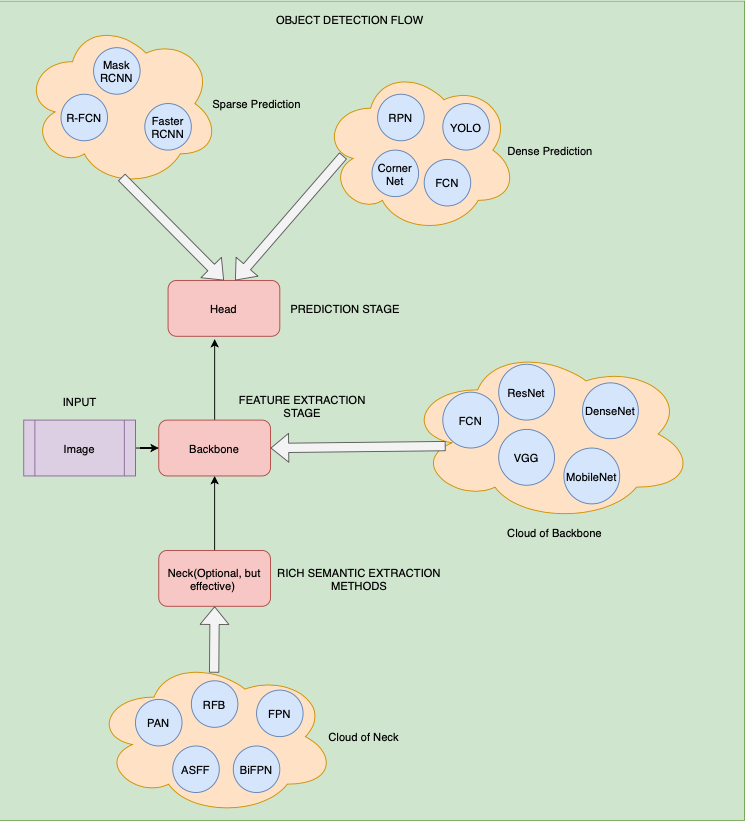
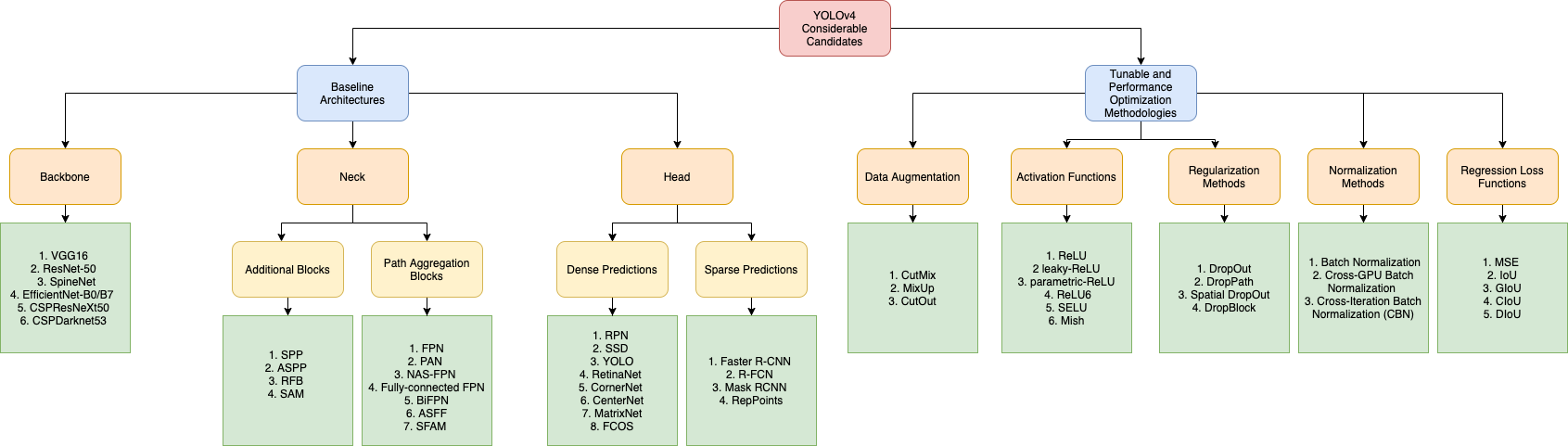
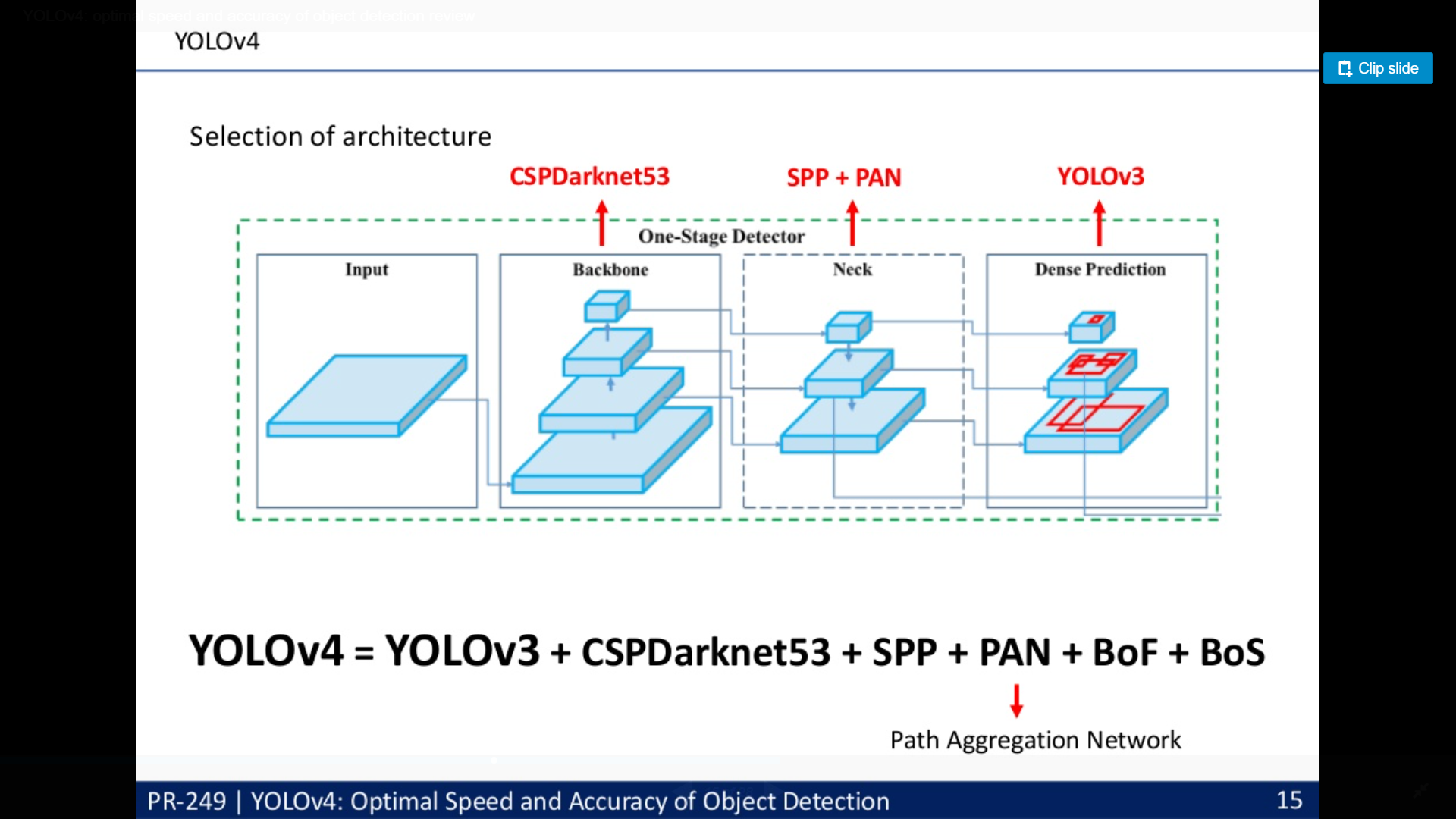
**STUDY ON YOLOv4 Architecture (Part2):**

Object detection flow:



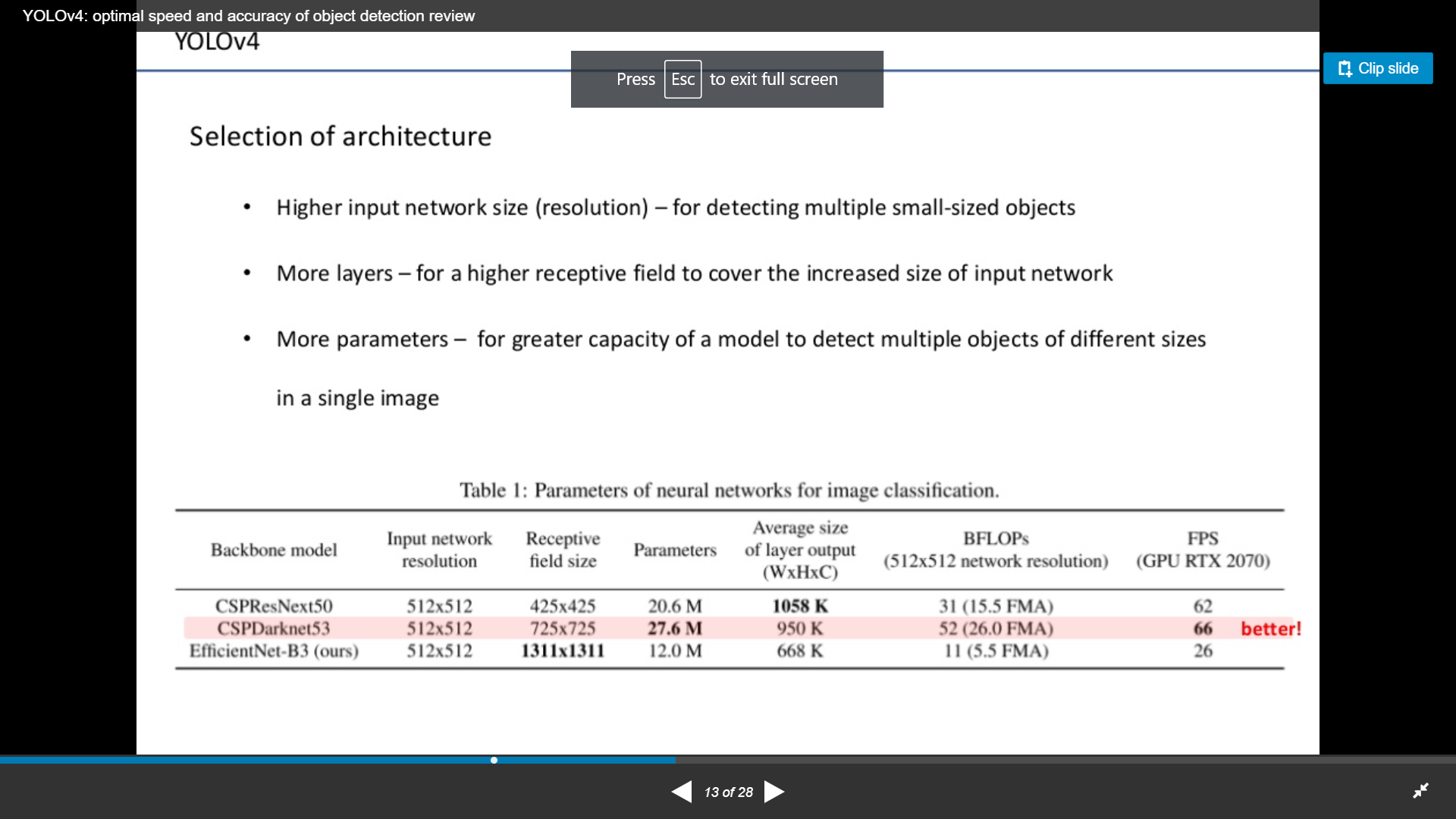




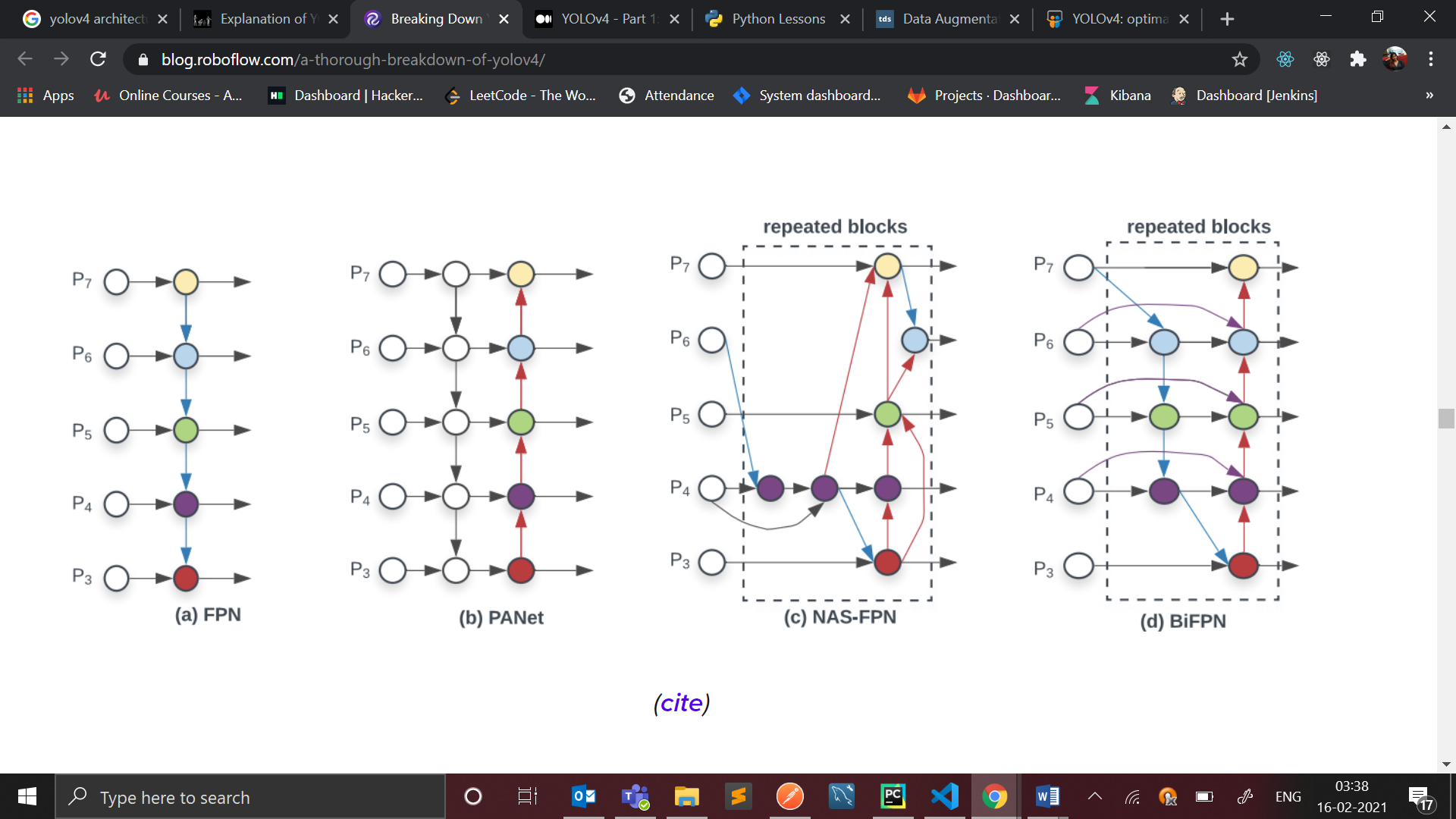
1. The authors considered the following backbones for the YOLOv4 object detector:

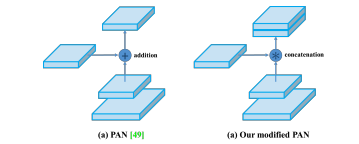
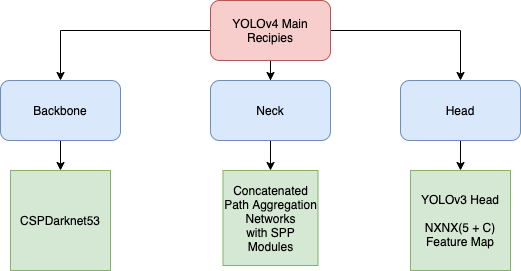
* CSPResNext50
* CSPDarknet53
* EfficientNet-B3

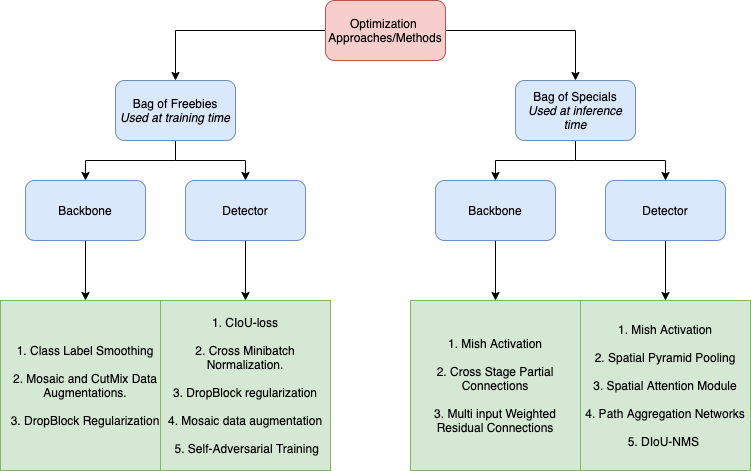
1. The CSPResNext50 and the CSPDarknet53 are both based on DenseNet.
2. DenseNet was designed to connect layers in convolutional neural networks with the following motivations:
   * to alleviate the vanishing gradient problem (it is hard to backprop loss signals through a very deep network),
   * to bolster feature propagation,
   * encourage the network to reuse features, and
   * reduce the number of network parameters.



1. In CSPResNext50 and CSPDarknet53, the DenseNet has been edited to separate the feature map of the base layer by copying it and sending one copy through the dense block and sending another straight on to the next stage.
2. The idea with the CSPResNext50 and CSPDarknet53 is to remove computational bottlenecks in the DenseNet and improve learning by passing on an unedited version of the feature map.
3. The next step in object detection is to mix and combine the features formed in the ConvNet backbone to prepare for the detection step. YOLOv4 considers a few options for the neck including:
   * FPN
   * PAN
   * NAS-FPN
   * BiFPN
   * ASFF
   * SFAM
4. The components of the neck typically flow up and down among layers and connect only the few layers at the end of the convolutional network.



1. Neck is leveraged in backbones for the extraction of rich semantic features that are further used for accurate predictions. One important benchmark for this is the receptive field.
2. After optimal testings, Spatial Pyramid Pooling(SPP) was tightly coupled with CSPDarknet53 which helped in the drastic increase of the receptive field and Modified Path Aggregation Networks for pyramidal structure instead of Feature Pyramid Networks used in YOLO4.
3. 
4. YOLOv4 deploys the same YOLO head as YOLOv3 for detection with the anchor based detection steps, and three levels of detection granularity.
5. YOLOv4 Head is taken into consideration for loss propagation and predictions.
6. 
7. Apart from different selective approaches in architecture designs, the authors also added two new “Bags” or optimization procedures to be used at the time of training and inference. These are stated as Bag of Freebies(BoG) and Bag of Specials(BoS).



1. Bag of Freebies:

Related Work
PR-249 | YOLOv4: Optimal Speed and Accuracy of Object Detection 10
Bag of Freebies (pre-processing + training...

1. Bag of Freebies was filled with different approaches used for both, backbone and detector modules in YOLOv4.

— Backbone

* + CutMix and Mosaic Augmentations helping the detector to learn different types of distribution of a given image under challenging circumstances such as occlusion, noise, etc.
  + Drop Block Regularization¹⁵ for learning spatially discriminating features.
  + Class Label Smoothing for better generalization on a dataset.

— Detector

* + CIoU Loss¹⁶ for better convergence on a bounding box regression.
  + Cross mini Batch Normalization¹ for collecting statistics inside the entire batch, instead of collecting statistics inside a single mini-batch.
  + Self Adversarial Training and Mosaic Augmentations for making them forbidden by adversarial attacks on a CNN.
  + Grid Sensitivity Elimination for solving the problem of the undetectable objects in an image.
  + Multiple Anchors for a particular ground truth for better regression stability
  + Cosine Annealing Scheduler for adjustment of the learning rate for sinusoidal training.
  + Random Anchor Shapes for consideration of generalized spatial sizes of the objects in an image.

1. Bag of Specials:

Related Work
PR-249 | YOLOv4: Optimal Speed and Accuracy of Object Detection
Bag of Specials (plugin modules + post-proces...

1. Plugin modules are the list of attributes or algorithms to increase the receptive field, strengthening feature integration capability, etc. whereas post-processing modules are used to filter out detections predicted by a detector. BoS for backbone and detector are mentioned below.

— Backbone

* + Mish Activation¹⁷ used in the final inference version which helped in getting a ~1% increase in Top 1 accuracy.
  + Cross Stage Partial Connections¹¹ for reducing total multiplications with convolutional filters/reducing time complexity.
  + Multi-input weighted residual connections (MiWRC).

— Detector

* + Mish Activation
  + Self Attention Module(SAM)¹⁸ for capturing global dependencies and tight input training.
  + Spatial Pyramid Pooling(SPP)¹² for increasing the receptive field of the overall network.
  + Path Aggregation Networks(PAN)¹³ for the better concatenation of local textures and global features of an object achieving better semanticity and accuracy.
  + DIoU NMS¹⁶ to remove the present confidence degradation and occlusion problems of standard NMS procedures.

Ref.:

<https://becominghuman.ai/explaining-yolov4-a-one-stage-detector-cdac0826cbd7#:~:text=YoloV4%20is%20an%20important%20improvement,network%20on%20a%20single%20GPU>.

<https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/>

<https://medium.com/visionwizard/yolov4-version-0-introduction-90514b413ccf>

<https://towardsdatascience.com/data-augmentation-in-yolov4-c16bd22b2617>

<https://www.slideshare.net/HoseongLee6/yolov4-optimal-speed-and-accuracy-of-object-detection-review>