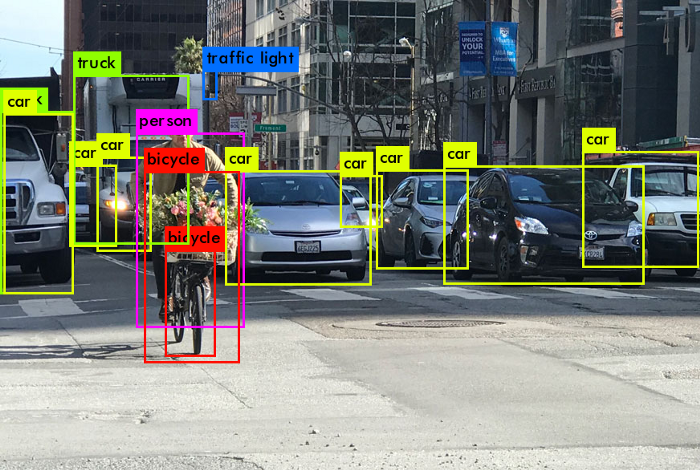
**YOLOv4: Optimal Speed and Accuracy of Object Detection**

-Saumya Arora



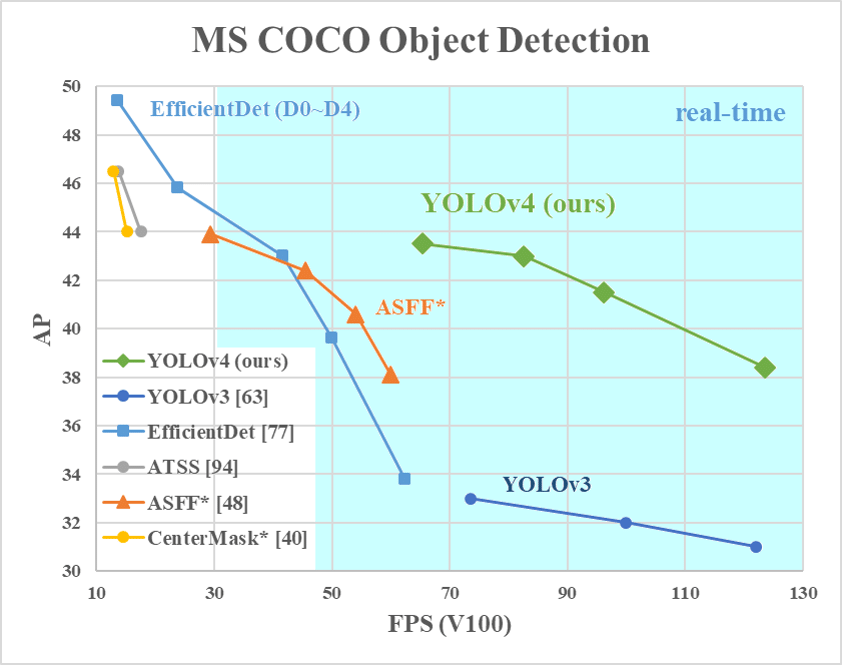
**Abstract**

There are a huge number of features that are said to improve Convolutional Neural Network (CNN) accuracy. Practical testing of combinations of such features on large datasets, and theoretical justification of the result, is required. Some features operate on certain models exclusively and for certain problems exclusively, or only for small-scale datasets; while some features, such as batch-normalization and residual connections, apply to the majority of models, tasks, and datasets. We assume that such universal features include Weighted-Residual-Connections (WRC), Cross-Stage-Partial-connections (CSP), Cross Mini-Batch Normalization (CmBN), Self-adversarial-training (SAT), and Mish-activation.

**Table of contents**

1. **INTRODUCTION**
2. **RELATED WORK**
3. **Object Detection Model**
4. **Parts of Object Detector**
5. **Bag of Freebies**
6. **Bag of Specials**
7. **METHODOLOGY**
8. **Selection of Architecture**
9. **Selection of BoF and BoS**
10. **Additional Improvements**
11. **YOLOv4**
12. **Uses of YOLOv4**
13. **ADVANTAGES**
14. **DISADVANTAGES**
15. **CONCLUSION**
16. **Introduction**

The majority of CNN-based object detectors are largely applicable only for recommendation systems**. For example, searching for free parking spaces via urban video cameras is executed by slow accurate models, whereas car collision warning is related to fast inaccurate models.** Improving the real-time object detection accuracy enables using them not only for hint generating recommendation systems but also for stand-alone process management and human input reduction. **Real-time object detector operation on conventional Graphics Processing Units (GPU) allows their mass usage at an affordable price. The most accurate modern neural networks do not operate in real-time and require a large number of GPUs for training with a large mini-batch size.** We address such problems by creating a CNN that operates in real-time on a conventional GPU, and for which training requires only one conventional GPU.

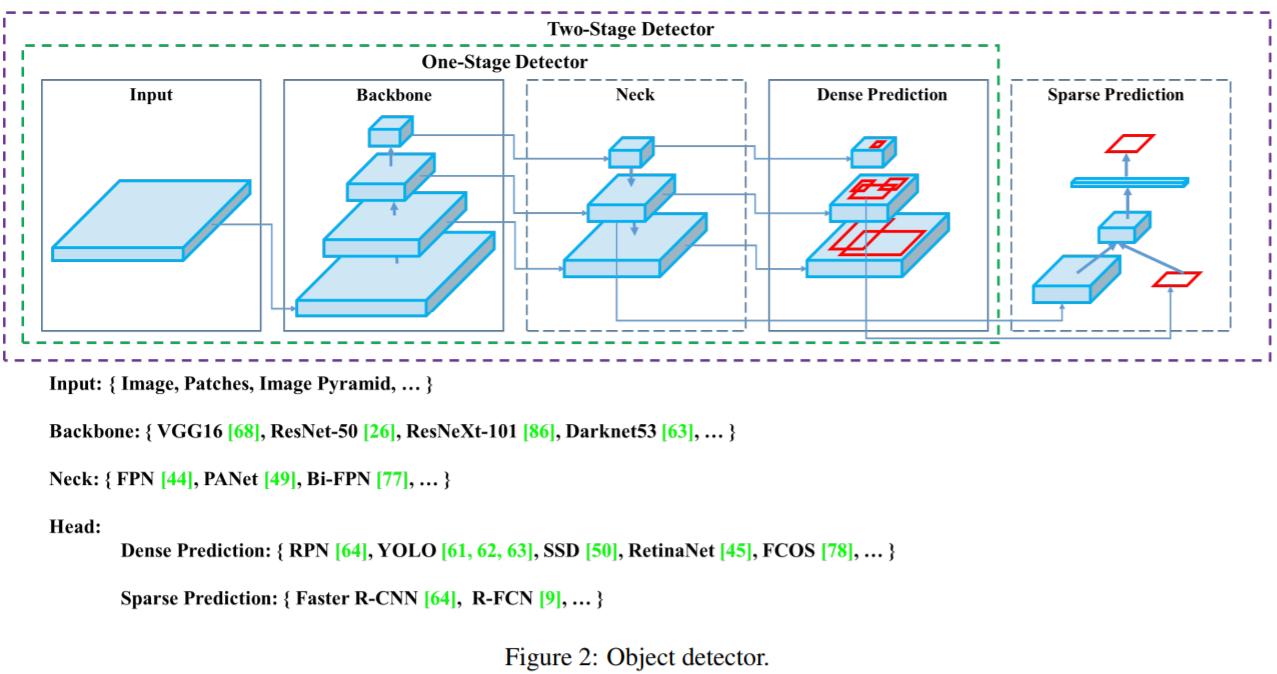
**Figure 1: Comparison of the proposed YOLOv4 and other state-of-the-art object detectors. YOLOv4 runs twice faster than efficient with comparable performance. Improves YOLOv3’s AP and FPS by 10% and 12%, respectively.**

The main goal of this work is designing a fast-operating speed of an object detector in production systems and optimization for parallel computations, rather than the low computation volume theoretical indicator. For example, anyone who uses a conventional GPU to train and test can achieve real-time, high quality, and convincing object detection results, as the YOLOv4 results are shown in Figure 1. Our contributions are summarized as follows:

**1.**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.

**2.**We verify the influence of state-of-the-art Bag-of-Freebies and Bag-of-Special methods of object detection during the detector training.

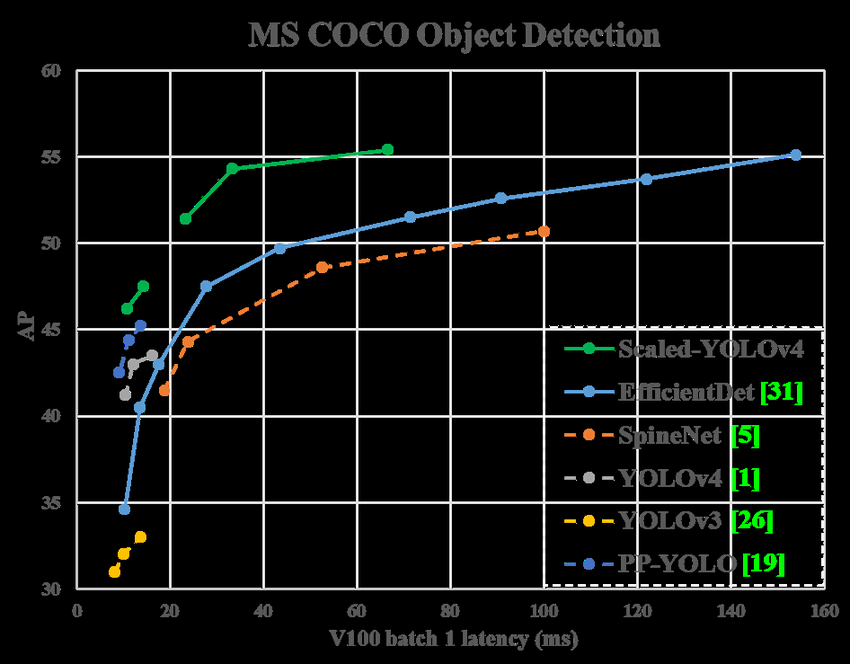
**3.**We modify state-of-the-art methods and make them more efficient and suitable for single GPU training, including CBN [89], PAN [49], SAM [85], etc



1. **Related work**
2. **Object Detection Models**

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. For those detectors running on GPU platform, their backbone could be VGG [68], ResNet [26], ResNeXt [86], or DenseNet [30]. For those detectors running on the CPU platform, their backbone could be Squeeze Net [31], Mobile Net [28, 66, 27, 74], or Shuffle Net [97, 53]. As to the head part, it is usually categorized into two kinds**, i.e., one-stage object detector and two-stage object detector**. **The most representative two-stage object detector is the R-CNN [19] series, including fast R-CNN [18], faster R-CNN [64], R-FCN [9], and Libra R-CNN [58].** Object detectors developed in recent years often insert some layers between the backbone and head, and these layers are usually used to collect feature maps from different stages

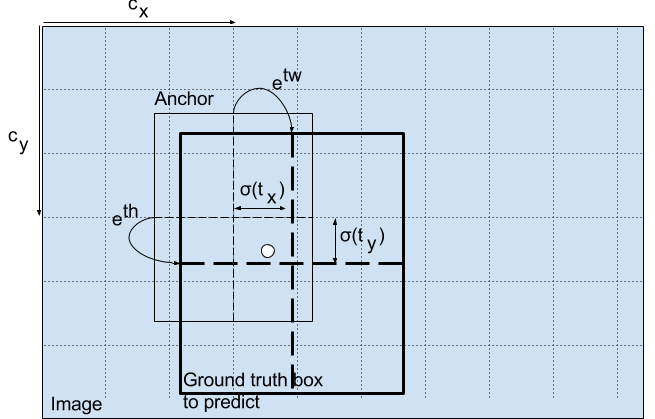
1. **Parts of Object Detector**
   1. **Input:** Image, Patches, Image Pyramid
   2. **Backbones:** VGG16 [68], ResNet-50 [26], SpineNet [12], EfficientNet-B0/B7 [75], CSPResNeXt50 [81], CSPDarknet53 [81]
   3. **Neck:**
2. **Additional blocks:** SPP [25], ASPP [5], RFB [47], SAM [85]
3. **Path-aggregation blocks:** FPN [44], PAN [49], NAS-FPN [17], Fully-connected FPN, BiFPN [77], ASFF [48], SFAM [98]
   1. **Heads:**
4. **Dense Prediction (one-stage):** RPN [64], SSD [50], YOLO [61], RetinaNet [45] (anchor based) ◦ CornerNet [37], CenterNet [13], MatrixNet [60], FCOS [78] (anchor free)
5. **Sparse Prediction (two-stage):** Faster R-CNN [64], R-FCN [9], Mask RCNN [23] (anchor based) ◦ RepPoints [87] (anchor free)

**Comparison of the proposed YOLOv4 and other state-of-the-art object detectors. The dashed line means only latency of model inference, while the solid line includes model inference and post-processing.**

1. **Bag of Freebies**

Usually, a conventional object detector is trained offline. **Therefore, researchers always like to take this advantage and develop better training methods that can make the object detector receive better accuracy without increasing the inference cost. We call these methods that only change the training strategy or only increase the training cost a “bag of freebies.”** The purpose of data augmentation is to increase the variability of the input images so that the designed object detection model has higher robustness to the images obtained from different environments.

**For example,** photometric distortions and geometric distortions are two commonly used data augmentation methods and they benefit the object detection task. In dealing with photometric distortion, we adjust the brightness, contrast, hue, saturation, and noise of an image. For geometric distortion, we add random scaling, cropping, flipping, and rotating.

**The data augmentation methods mentioned above are all pixel-wise adjustments, and all original pixel information in the adjusted area is retained. In addition, some researchers engaged in data augmentation put their emphasis on simulating object occlusion issues.** They have achieved good results in image classification and object detection In addition, some researchers have proposed the methods of using multiple images together to perform data augmentation. For example, MixUp [92] uses two images to multiply and superimpose with different coefficient ratios and then adjusts the label with these superimposed ratios. As for CutMix [91], it is to cover the cropped image to the rectangle region of other images, and adjusts the label according to the size of the mixing area. In addition to the above-mentioned methods, style transfer GAN [15] is also used for data augmentation, and such usage can effectively reduce the texture bias learned by CNN.



In dealing with the problem of semantic distribution bias, a very important issue is that there is a problem of **data imbalance between different classes**, and this problem is **often solved by hard negative example mining** [72] or online hard example mining [67] in two-stage object detector.

**But the example mining method does not apply to a one-stage object detector,** because this kind of detector belongs to the dense prediction architecture. Therefore **Lin et al.** [45] proposed **focal loss to deal with the problem of data imbalance existing between various classes.** Another very important issue is that it is **difficult to express the relationship of the degree of association between different categories with the one-hot hard representation.**



To make this issue processed better, some researchers recently **proposed IoU loss** [90], which puts the coverage of **predicted BBox area and ground truth BBox area into consideration.** The IoU loss computing process **will trigger the calculation of the four coordinate points of the BBox by executing IoU with the ground truth and then connecting the generated results into a whole code.** Because IoU is a scale-invariant representation. Recently, some researchers have continued to improve IoU loss.

**For example**, GIoU loss [65] is to includes the shape and orientation of the object in addition to the coverage area. They proposed to find the smallest area BBox that can simultaneously cover the predicted BBox and ground truth BBox, and use this BBox as the denominator to replace the denominator originally used in IoU loss. As for DIoU loss [99], it additionally considers the distance of the center of an object, and CIoU loss [99], on the other hand simultaneously considers the overlapping area, the distance between center points, and the aspect ratio. CIoU can achieve better convergence speed and accuracy on the BBox regression problem.

1. **Bag of Specials**

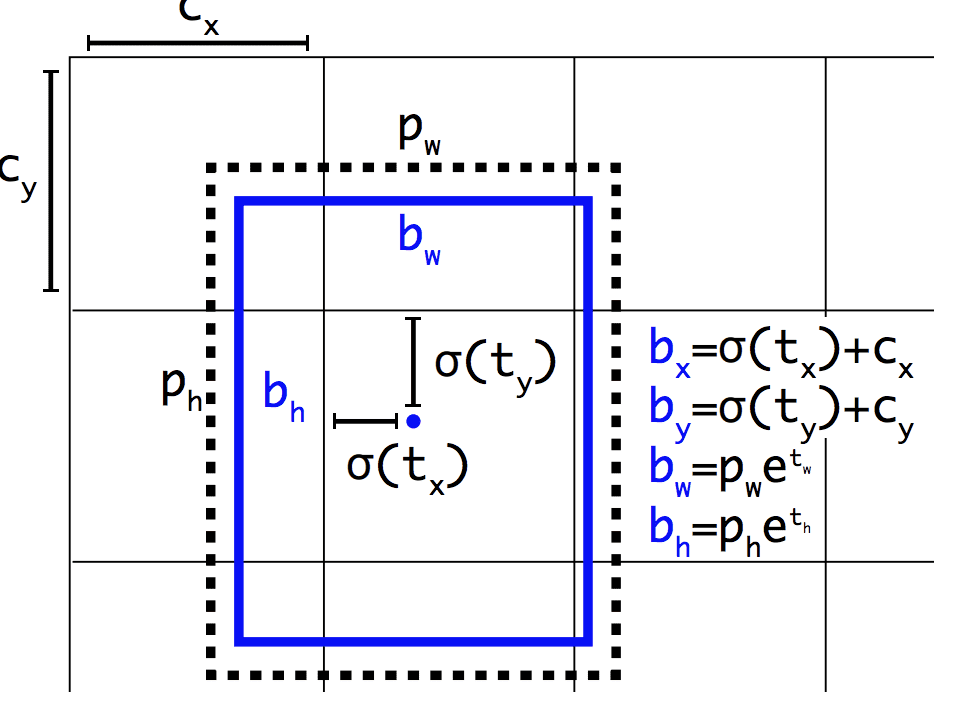
**For those plugin modules and post-processing methods that only increase the inference cost by a small amount but can significantly improve the accuracy of object detection, we call them “bag of specials”.** Generally speaking, these plugin modules are for enhancing certain attributes in a model, such as enlarging receptive field, introducing attention mechanism, or strengthening feature integration capability, etc., and post-processing is a method for screening model prediction results.

**Common modules that can be used to enhance receptive fields are SPP [25], ASPP [5], and RFB [47].** Thus, in the design of YOLOv3 [63], Redmon and Farhadi improve the SPP module to the concatenation of max-pooling outputs with kernel **size k × k, where k = {1, 5, 9, 13}, and stride equals 1.**

Under this design, a relatively large k × k max-pooling effectively increases the receptive field of the backbone feature. After adding the improved version of the SPP module, YOLOv3-608 upgrades AP50 by 2.7% on the MS COCO object detection task at the cost of 0.5% extra computation**.**

**The difference in operation between ASPP [5] module and the improved SPP module is mainly from the original k×k kernel size, max-pooling of stride equals to 1 to several 3 × 3 kernel size, dilated ratio equals to k, and stride equal to 1 in dilated convolution operation.** RFB module is to use several dilated convolutions of k×k kernel, dilated ratio equals to k, and stride equals to 1 to obtain a more comprehensive spatial coverage than ASPP. RFB [47] only costs 7% extra inference time to increase the AP50 of SSD on MS COCO by 5.7%.

The attention module that is often used in object detection is mainly divided into channel-wise attention and pointwise attention, and the representatives of these **two attention models are Squeeze-and-Excitation (SE) [29] and Spatial Attention Module (SAM) [85], respectively.** Although the SE module can improve the power of ResNet50 in the ImageNet image classification task 1% top-1 accuracy at the cost of only increasing the computational effort by 2%, on a GPU usually it will increase the inference time by about 10%, so it is more appropriate to be used in mobile devices. But for SAM, it only needs to pay 0.1% extra calculation and it can improve ResNet50-SE 0.5% top-1 accuracy on the ImageNet image classification task. Best of all, it does not affect the speed of inference on the GPU at all.

**Since multi-scale prediction methods such as FPN have become popular, many lightweight modules that integrate different feature pyramids have been proposed. The modules of this sort include SFAM [98], ASFF [48], and BiFPN [77]. The main idea of SFAM is to use the SE module to execute channel-wise level re-weighting on multi-scale concatenated feature maps.** **As for ASFF, it uses SoftMax as point-wise level reweighting and then adds feature maps of different scales. In BiFPN, the multi-input weighted residual connections are proposed to execute scale-wise level re-weighting, and then add feature maps of different scales.**

**Example of the Representation Chosen when Predicting Bounding Box Position and Shape Taken from YOLO9000: Better, Faster, Stronger**

**The post-processing method commonly used in deep learning-based object detection is NMS, which can be used to filter those BBoxes that badly predict the same object, and only retain the candidate BBoxes with a higher response.** The way NMS tries to improve is consistent with the method of optimizing an objective function. The original method proposed by NMS does not consider the context information, so **Girshick et al.** [19] added classification confidence score in R-CNN as a reference, and according to the order of confidence score, greedy NMS was performed in the order of high score to the low score. **The DIoU NMS [99] developers’ way of thinking is to add the information of the center point distance to the BBox screening process based on soft NMS. It is worth mentioning that, since none of the above postprocessing methods directly refer to the captured image features, post-processing is no longer required in the subsequent development of an anchor-free method.**

1. **Advantages**
2. It’s incredibly fast and can process 45 frames per second to 150 frames per second.
3. YOLO also understands generalized object representation.
4. The network can generalize the image better.
5. **Disadvantages**
6. Comparatively low recall and more localization error compared to Faster R\_CNN.
7. Struggles to detect close objects because each grid can propose only 2 bounding boxes.
8. Struggles to detect small objects.
9. **Conclusion**

We offer a state-of-the-art detector that is faster (FPS) and more accurate (MS COCO AP50...95 and AP50) than all available alternative detectors. The detector described can be trained and used on a conventional GPU with 8-16 GB-VRAM this makes its broad use possible. The original concept of one-stage anchor-based detectors has proven its viability. We have verified a large number of features and selected them for use such them for improving the accuracy of both the classifier and the detector. These features can be used as best-practice for future studies and developments.

**REFERENCE AND LINKS**

**MOST OF THE CONTENT IS FROM THE RESEARCH PAPER WHICH HAS BEEN PROVIDED TO US.**

**[ ] THESE BOXES CONTAINS THE LINK NUMBER, WHICH HAS BEEN PROVIDED IN THE RESEARCH PAPER**

**LINKS OTHER THAN RESEARCH PAPER:**

<https://www.google.com/search?q=Figure+1:+Comparison+of+the+proposed+YOLOv4+and+other+state-of-the-art+object+detectors.+YOLOv4+runs+twice+faster+than+efficient+with+comparable+performance.+Improves+YOLOv3%E2%80%99s+AP+and+FPS+by+10%25+and+12%25,+respectively&rlz=1C1JZAP_enIN871IN871&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiy56bjnLnxAhUOzTgGHYgECF0Q_AUoAnoECAEQBQ&biw=1536&bih=754&dpr=1.25#imgrc=MCtWVsw30NF7mM>

<https://www.google.com/search?q=yolo+prediction&rlz=1C1JZAP_enIN871IN871&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjjqqXDuLnxAhV573MBHRFMC1YQ_AUoA3oECAEQBg&biw=1536&bih=754#imgrc=-iUfGHNlOy6sdM>

<https://www.google.com/search?q=yolo+prediction&rlz=1C1JZAP_enIN871IN871&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjjqqXDuLnxAhV573MBHRFMC1YQ_AUoA3oECAEQBg&biw=1536&bih=754#imgrc=-iUfGHNlOy6sdM> <https://www.google.com/search?q=problem+faced&tbm=isch&ved=2ahUKEwik-8XEuLnxAhXHs0sFHQJ5A1kQ2-cCegQIABAA&oq=problem+faced&gs_lcp=CgNpbWcQAzICCAAyAggAMgIIADICCAAyAggAMgIIADICCAAyAggAMgIIADICCAA6BAgAEBg6BAgAEEM6BQgAELEDOggIABCxAxCDAToHCAAQsQMQQzoGCAAQChAYULDhPFiYkj1gm5Y9aAFwAHgFgAHdBIgBrymSAQwwLjQuMTAuMi4xLjKYAQCgAQGqAQtnd3Mtd2l6LWltZ7ABAMABAQ&sclient=img&ei=a0nZYOSZMsfnrtoPgvKNyAU&bih=754&biw=1536&rlz=1C1JZAP_enIN871IN871#imgrc=kLn0-68KOFTyHM&imgdii=JBO4g2uYblWYXM>

<https://www.google.com/search?q=solution&tbm=isch&ved=2ahUKEwik-8XEuLnxAhXHs0sFHQJ5A1kQ2-cCegQIABAA&oq=solution&gs_lcp=CgNpbWcQAzIFCAAQsQMyBQgAELEDMgIIADICCAAyAggAMgIIADICCAAyAggAMgIIADICCAA6BAgAEBg6CAgAELEDEIMBOgQIABBDUPu4TViH6k1g9e5NaAJwAHgEgAGJA4gBthySAQgwLjEuMTIuMpgBAKABAaoBC2d3cy13aXotaW1nsAEAwAEB&sclient=img&ei=a0nZYOSZMsfnrtoPgvKNyAU&bih=754&biw=1536&rlz=1C1JZAP_enIN871IN871#imgrc=9Dc-4F0X8-tlKM>

<https://www.researchgate.net/figure/Parameters-of-neural-networks-for-image-classification_tbl1_340883401>

<https://www.google.com/search?q=advantages+of+yolo+predictor&rlz=1C1JZAP_enIN871IN871&oq=advantages+of+yolo+predictor&aqs=chrome..69i57j33i10i160.8915j0j15&sourceid=chrome&ie=UTF-8>

<https://www.geeksforgeeks.org/yolo-you-only-look-once-real-time-object-detection/>

**GITHUB**

<https://github.com/saumyaarora80/YOLO-Predictor>