**Compare three different Yolo Versions V3, V5, and the latest version:**

| **Models** | **YOLOv3** | | **YOLOv5** | | **Latest YOLOv.** |
| --- | --- | --- | --- | --- | --- |
| **[5 pt] Comparison between these Models.** | | | | | |
| **Deploying year** | **2018** | | **2020** | | **2023 (Yolo v8)** |
| **Architecture** | * **The input image is first resized to 416x416 pixels** * **53 conv-layers(darknet-53)** * **FPN to enhance the feature maps** | | * **The backbone network is a modified version of the CSPDarknet53 architecture,** * **The YOLOv5 architecture includes several key optimizations that make it faster and more accurate than previous versions of YOLO. These optimizations include:** * **Dynamic anchor assignment: YOLOv5 dynamically assigns anchor boxes to different scales based on the input image size, which improves detection accuracy for objects of different sizes.** * **Improved focal loss: YOLOv5 uses an improved version of the focal loss function, which helps to reduce false positives and improve detection accuracy.** * **Feature pyramid network (FPN): YOLOv5 includes an FPN that helps to capture features at different scales and improves detection accuracy for small objects.** * **Scaled-YOLO: YOLOv5 uses a scaled version of the model that can be trained on larger input image sizes, which improves detection accuracy and reduces false positives.** | | **This is for Yolov7, there isn’t a scientific paper of Yolov8 yet** |
| **Prediction scales** | **Scale 1: This scale corresponds to the highest level in the feature pyramid and has the lowest resolution. Objects that are relatively large and have a low aspect ratio are detected at this scale.**  **Scale 2: This scale corresponds to an intermediate level in the feature pyramid and has a higher resolution than Scale 1. Objects that are medium-sized and have a moderate aspect ratio are detected at this scale.**  **Scale 3: This scale corresponds to the lowest level in the feature pyramid and has the highest resolution. Objects that are relatively small and have a high aspect ratio are detected at this scale.** | | **13x13 grid: This feature map is associated with the smallest anchor boxes and is used to detect small objects. Each grid cell predicts 3 anchor boxes.**  **26x26 grid: This feature map is associated with medium-sized anchor boxes and is used to detect medium-sized objects. Each grid cell predicts 3 anchor boxes.**  **52x52 grid: This feature map is associated with the largest anchor boxes and is used to detect large objects. Each grid cell predicts 3 anchor boxes.** | | YOLO v7, like many object detection algorithms, struggles to detect small objects. It might fail to accurately detecting objects in crowded scenes or when objects are far away from the camera.  YOLO v7 is also not perfect at detecting objects at different scales. This can make it difficult to detect objects that are either very large or very small compared to the other objects in the scene.  YOLO v7 can be sensitive to changes in lighting or other environmental conditions, so it may be inconvenient to use in real-world applications where lighting conditions may vary.  YOLO v7 can be computationally intensive, which can make it difficult to run in real-time on resource-constrained devices like smartphones or other edge devices. |
| **Number of Anchors** | **9 anchor boxes for each grid cell (608\*608)** | | **9 anchor boxes for each grid cell** | | **9 anchor boxes (v7)** |
| **Number of params** | **8,861,918 parameters** | | **21 million parameters (yolov5m)**  **here are also larger variants of YOLOv4 that have more parameters. For example, the YOLOv4x model has around 170 million parameters.** | | **The YOLOv7 normal model with almost 37 million parameters** |
| **Loss Function** |  | | Binary cross entropy for the objectness and classification scores,  Box-per-cell level prediction instead of cell level prediction for the class probabilities, so a slightly different penalization for the classification terms,  CIoU Loss instead of MSE for the regression terms (x,y,w,h). CIoU stands for Complete Intersection over Union, and is not so far from the MSE loss. It proposes to compare width and height a bit more interestingly (consistency between aspect ratios), but it keeps the MSE for the comparison between bounding box centers | | **The loss function of the YOLOv7 model consists of three parts: localization loss (Lbox), confidence loss (Lobj), and classification loss (Lcls). The total loss is the weighted sum of the three losses1. All losses are mean-squared errors, except classification loss, which uses cross-entropy function** |
| **[5 pt ]Test video on the previous models.** | | | | |  |
| **Number of objects** | **2614** | | **2583** | | **2483** |
| **speed(Detection Time)**  **Cpu or GPU** | **6597.63(GPU)** | | **2269.85(GPU)** | | **1600.63(GPU)** |
| **[2 pt] Bonus: Repeat the comparison for the next models.** | | | | | |
| **Models** | **yolov5n** | **yolov5s** | **yolov5m** | **yolov5l** | **yolov5x** |
|
| **Number of objects** | **2272** | **2428** | **2583** | **2668** | **2719** |
| **speed(Detection Time)**  **Cpu or GPU** | **1340.86(GPU)** | **1962.17(GPU)** | **2585.23(GPU)** | **3938.80(GPU)** | **6312.56(GPU)** |