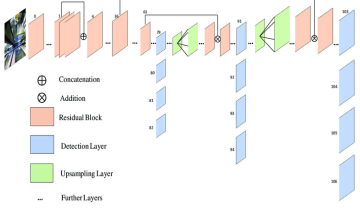
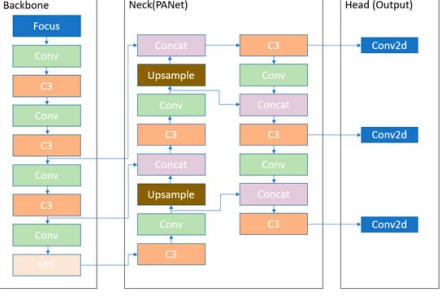
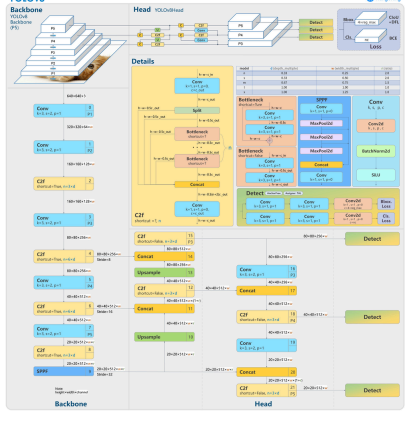
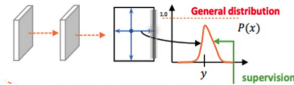
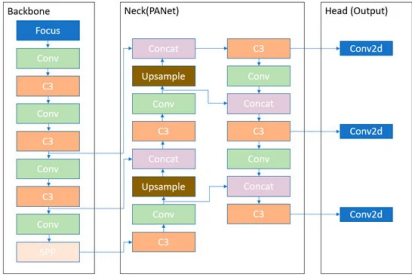


Compare three different Yolo Versions V3, V5, and the latest version:

Models	YOLOv3	YOLOv5	Latest YOLOv.
[5 pt] Comparison between these Models.			
Deploying year	2018	2020	Jan. 2023
Architecture	 <p>The diagram shows the YOLOv3 architecture. It starts with an input image, followed by a series of convolutional layers (C1, C2, C3, C4, C5) and max pooling layers (P1, P2, P3). The output is then processed by three parallel detection heads (D1, D2, D3) at different scales. A legend indicates: Concatenation (⊕), Addition (+), Residual Block (orange box), Detection Layer (blue box), Upsampling Layer (green box), and Further Layers (dashed line).</p>	 <p>The diagram shows the YOLOv5 architecture. It is divided into three main parts: Backbone, Neck (PANet), and Head (Output). The Backbone consists of a Focus layer followed by a series of C3 (Convolutional) and Conv (Convolutional) layers. The Neck (PANet) consists of a series of Concat (Concatenation) and Upsample layers. The Head (Output) consists of a series of Conv2d (Convolutional) layers.</p>	 <p>The diagram shows the YOLOv8 architecture. It is divided into three main parts: Backbone, Neck, and Head. The Backbone consists of a series of Conv (Convolutional) and C2f (Convolutional) layers. The Neck consists of a series of Concat (Concatenation) and Upsample layers. The Head consists of a series of Conv (Convolutional) and C2f (Convolutional) layers. The diagram also includes a table of model parameters and a list of model variants.</p>
Prediction scales	YOLOv3u makes predictions at three different scales with strides 32, 16, and 8. Then, it combines the results to get the final detection.	YOLOV5n makes a four-scale prediction head. It is modified to capture objects with a large-scale variation. An additional prediction scale is added to the prediction head of the proposed detection model.	YOLOv8n predict mode can generate predictions for various tasks, returning either a list of Results objects or a memory-efficient generator of Results objects when using the streaming mode.

Number of Anchors	YOLOv3u in particular has 3 anchors.		YOLOv5n has 9 anchor boxes and the number of clusters is 9.		YOLOv8n is an anchor-free model. This means it predicts directly the center of an object instead of the offset from a known anchor box.
Number of params	YOLOv3u has 8,861,918 learnable parameters and 13 Convolution layers.		YOLOv5n model has 1.9 million different parameters		YOLOv8n has nearly 5 million parameters.
Loss Function	$\lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} \left[ (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$ $+ \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} \left[ \left( \sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left( \sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$ $+ \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2$ $+ \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2$ $+ \sum_{i=0}^{s^2} \mathbb{1}_i^{obj} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$		$LOSS = L_{classification} + L_{confidence} + L_{CIoU}$ $L_{classification} = \sum_{i=0}^{s^2} \ell_i^{obj} \sum_{j=0}^B \left[ \left( p_i(c) - \hat{p}_i(c) \right)^2 \right]$ $L_{confidence} = \sum_{i=0}^{s^2} \sum_{j=0}^B \ell_i^{obj} \left[ \left( C_i - \hat{C}_i \right)^2 \right] + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B \ell_i^{noobj} \left[ \left( C_i - \hat{C}_i \right)^2 \right]$		YOLOv8n uses VFL Loss as classification loss and DFL Loss+CIoU Loss as classification loss. $VFL(p, q) = \begin{cases} -q(q \log(p) + (1 - q) \log(1 - p)) & q > 0 \\ -\alpha p^\gamma \log(1 - p) & \text{CSDN @whaosoft} \end{cases}$  $DFL(S_i, S_{i+1}) = -((y_{i+1} - y) \log(S_i) + (y - y_{i+1}) \log(S_{i+1})).$
[5 pt ]Test video on the previous models.					
Number of objects	2614		2272		2483
speed(Detection Time)	597224.87 msec = 597 sec		26018.24 msec = 26 sec		28207.86 msec = 28 sec
Cpu					
[2 pt] Bonus: Repeat the comparison for the next models.					
Models	yolov5n	yolov5s	yolov5m	yolov5l	yolov5x

Deploying year	2020				
Architecture	 <p>The diagram illustrates the YOLOv5 architecture, divided into three main components: Backbone, Neck (PANet), and Head (Output).  <b>Backbone:</b> Starts with a 'Focus' layer, followed by a sequence of 'Conv' (green) and 'C3' (orange) blocks. The output of the backbone is fed into the Neck.  <b>Neck (PANet):</b> This section handles feature fusion and upsampling. It consists of two parallel paths. The left path includes 'Concat' (purple), 'Upsample' (brown), 'Conv' (green), and 'C3' (orange) blocks. The right path includes 'C3' (orange), 'Conv' (green), and 'Concat' (purple) blocks. Arrows indicate the flow of information between these blocks and the backbone.  <b>Head (Output):</b> The final output stage, consisting of three 'Conv2d' (blue) layers that produce the detection results.</p>				
Prediction scales	YOLOV5 makes a four-scale prediction head. It is modified to capture objects with a large-scale variation. An additional prediction scale is added to the prediction head of the proposed detection model.				
Number of Anchors	YOLOv5n has 9 anchor boxes and the number of clusters is 9.				
Number of params	YOLOv5n model has 1.9 million different parameters.	YOLOv5s model has 7.2 million different parameters.	YOLOv5m model has 21.2 million different parameters.	YOLOv5l model has 46.5 million different parameters.	YOLOv5x model has 86.7 million different parameters

Loss Function	$LOSS = L_{classification} + L_{confidence} + L_{CIoU}$ $L_{classification} = \sum_{i=0}^{s^2} \ell_i^{obj} \sum_{j=0}^B \left[ \left( p_i(c) - \hat{p}_i(c) \right)^2 \right]$ $L_{confidence} = \sum_{i=0}^{s^2} \sum_{j=0}^B \ell_i^{obj} \left[ \left( C_i - \hat{C}_i \right)^2 \right] + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^B \ell_i^{noobj} \left[ \left( C_i - \hat{C}_i \right)^2 \right]$				
Test video on the previous models.					
Number of objects	2272	2428	2583	2668	2719
speed(Detection Time)	26217.75 msec = 26 sec	63514.08 msec = 63 sec	154673.03 msec = 154 sec	300879.21 msec = 300 sec	535098.98 msec = 535 sec
Cpu					

### **Assumptions:**

For the comparison, I have used Yolo 3u, 5n, and 8n.

### **Referefnecce:**

- [1] <https://blog.roboflow.com/yolov5-improvements-and-evaluation/>
- [2] <https://blog.roboflow.com/whats-new-in-yolov8/>
- [3] <https://www.mdpi.com/2077-1312/10/9/1230#:~:text=To%20suppress%20the%20effect%20of%20the%20proposed%20detection%20model.>
- [4] [https://www.coursera.org/learn/programming-languages?fbclid=IwAR1ao8Y4v6DIiZROziiU\\_nX\\_XRVt6qQdkfcLGNP3Oicp9uGq2ICYJkg5Ulo](https://www.coursera.org/learn/programming-languages?fbclid=IwAR1ao8Y4v6DIiZROziiU_nX_XRVt6qQdkfcLGNP3Oicp9uGq2ICYJkg5Ulo)
- [5] [https://blog.csdn.net/qq\\_29788741/article/details/128626422](https://blog.csdn.net/qq_29788741/article/details/128626422)
- [6] <https://github.com/ultralytics/yolov5>

