Fine-Tuning YOLOv8 for horses pose-estimation : Results

In this document, we will analyze the different outcomes of the YOLOv8 model trained for pose estimation on the horse10 dataset. Initially, we will examine the loss values for both training and validation datasets. Subsequently, we will investigate the precision/recall values and mAP50/mAP95 scores. In the second phase, we will study the F1-confidence curve. Following that, we will delve into the confusion matrix. Lastly, we will showcase some model predictions on new images, displaying the bounding boxes and pose estimation keypoints.

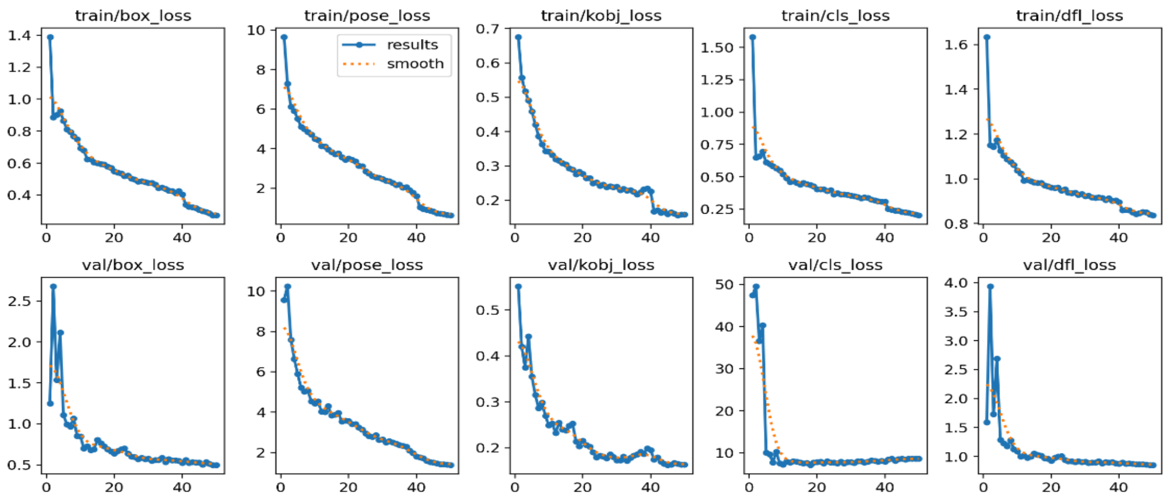
We trained our model for 50 epochs.

We have put all the necessary files required to train the model and execute the notebook in the following Google Drive link: https://drive.google.com/drive/folders/1d9fVKFxu7VuCoVFZafnwSBkWzrkn2_ap?usp=sharing

I- Loss Values:

→The loss values we will analyze are **box_loss, pose_loss, obj_loss, cls_loss, and dfl_loss** for both **training** and **validation** sets.

- box_loss: This represents the loss associated with the bounding box coordinates predicted by the model.
- pose_loss: This refers to the loss related to the predicted poses of the objects.
- obj_loss: This loss evaluates the model's ability to differentiate between regions containing objects and those without objects by assessing the accuracy of its objectness score predictions.
- cls_loss: This is the loss associated with the classification of objects into different classes. It evaluates the accuracy of the model in assigning the correct class labels to the detected objects.
- dfl_loss: This loss reflects any errors caused by special deformable layers or components within the model's design. It measures how much these unique parts contribute to mistakes in the model's predictions.



→We observe a **rapid decrease** in the values of the different train losses, stabilizing around epoch number **40**.Hence, the model did not require many epochs to converge during training on our dataset.

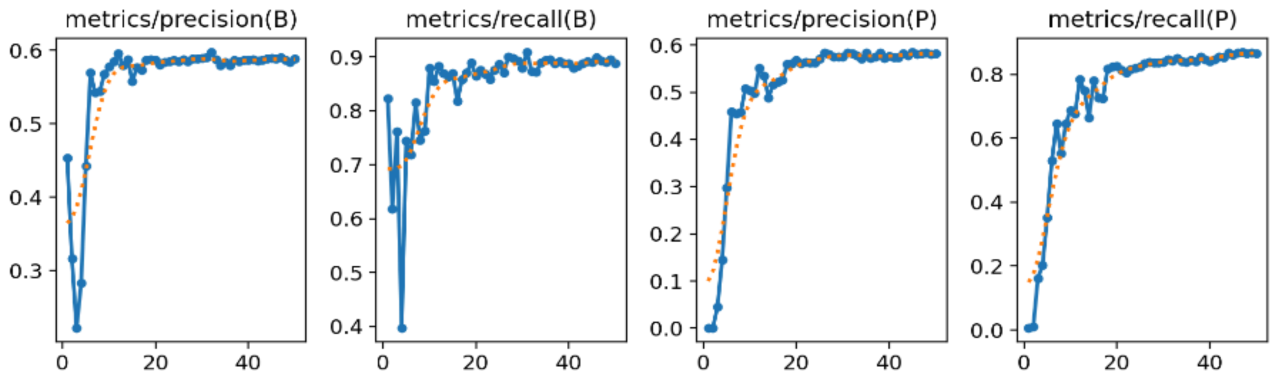
As for the corresponding validation loss values, they **also decrease rapidly**, with some **fluctuations** during the initial epochs. However, these fluctuations cease after epoch number **20**.

II- Precision/Recall Values:

-**Precision:** It is calculated as the number of true positive predictions divided by the total number of positive predictions made by the model. It tells us the proportion of **correctly predicted positive cases** among a**ll the cases predicted as positive** by the model → **"Of all the items the model predicted as positive, how many were actually positive?"**

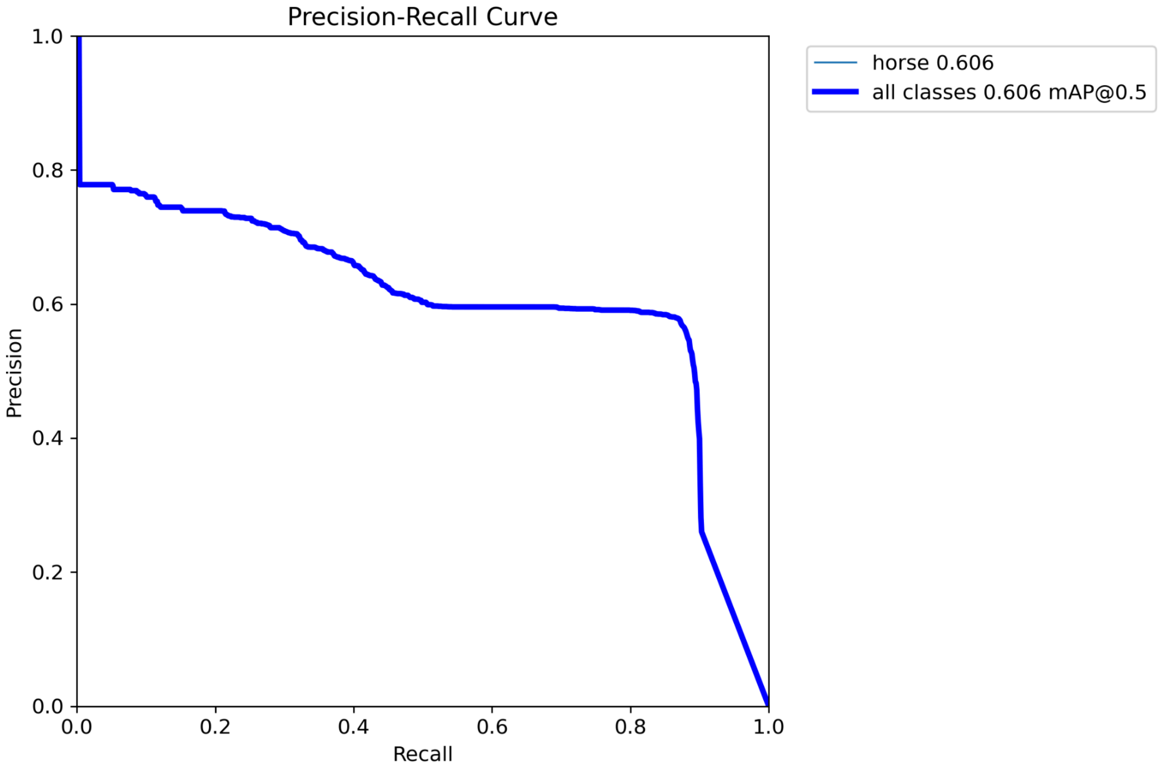
-**Recall:** It is calculated as the number of true positive predictions divided by the total number of actual positive cases in the dataset. It indicates the proportion of **correctly predicted positive cases** among all **the actual positive cases**.

→ **"Of all the actual positive cases, how many did the model manage to identify correctly?"**



→ B stands for “bounding-box” and P stands for “pose”.
→For Precision(B) and Recall(B), the values experience a brief decline in the early epochs, followed by a rapid rise, exhibiting fluctuations from epochs **10 to 40**, and eventually stabilizing from epoch **40 to 50** at 0.59 and 0.89. As for Precision(P) and Recall(P), they show a rapid ascent from the outset of training with fluctuations, then stabilize around epoch **40** at 0.59 and 0.85.
→The initial decline followed by a rapid rise (for Precision(B) and Recall(B)) indicates that the model initially struggles to accurately predict bounding boxes but quickly improves as training progresses.

→ Stabilization implies that the model has converged, and its performance regarding predictions has become consistent and reliable.

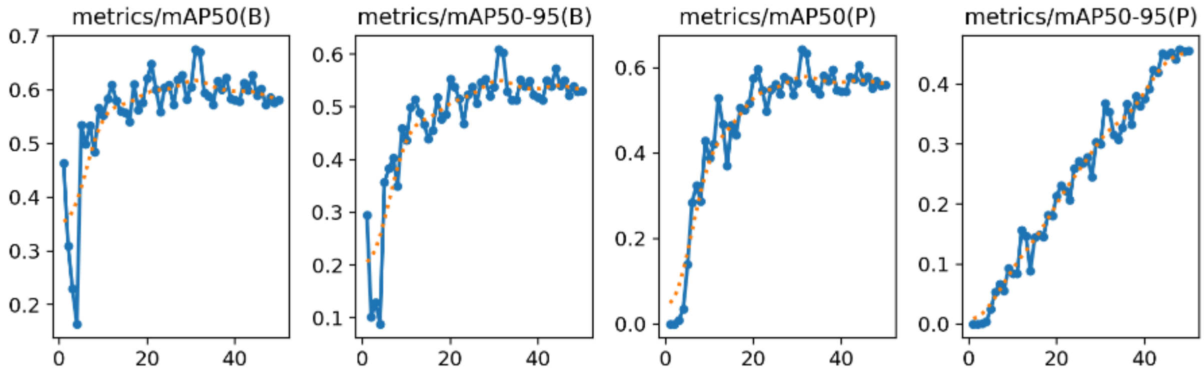


→The curve initiates at recall=0 with precision values of 1. Subsequently, as recall increases, precision undergoes a gradual decrease. Upon reaching recall=0.5, precision experiences a sharp decline from 0.6 to 0.
→These variations highlight the trade-off between precision and recall and provide insights into the model's strengths and weaknesses in identifying positive instances while minimizing false positives.

III- mAP50/mAP50-95 Values:

-In the task of HPE, the **Average Precision (AP)** metric based on OKS is widely used to evaluate algorithm performance. The AP metric calculates the average precision based on different thresholds and object sizes. For example, AP50 represents the average precision at an OKS threshold of 50: The "50" in **mAP50** refers to the threshold used for Intersection over Union (IoU), which is a measure of how much two bounding boxes overlap.

Similar to mAP50, **mAP50-95** calculates the average precision across different classes, but it considers a range of IoU thresholds from 0.5 to 0.95

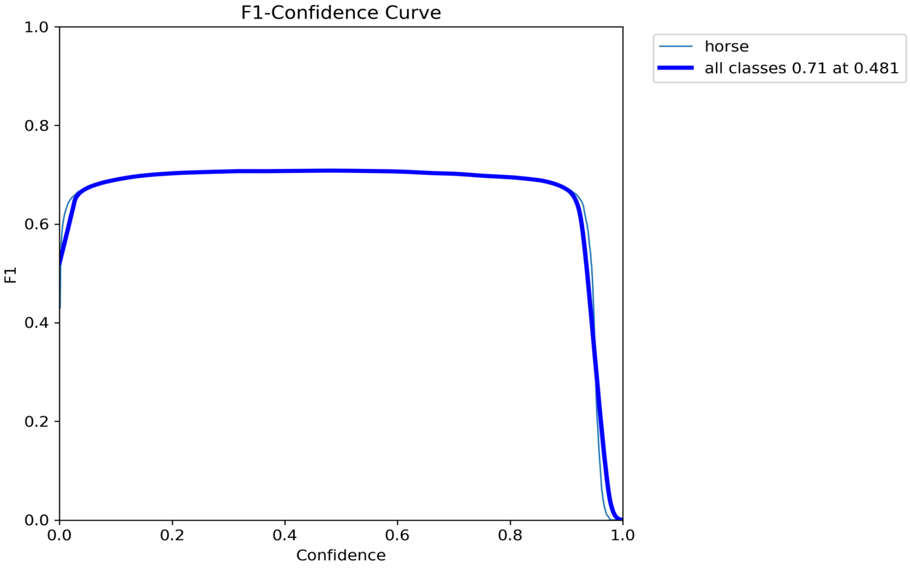


→For mAP50(B) and mAP50-95(B), the values experience a brief decline in the early epochs, followed by a rapid rise, exhibiting fluctuations from **epochs 10 to 45**.

As for mAP50(P) and mAP50-95(P), they show a rapid ascent from the outset of training with fluctuations.

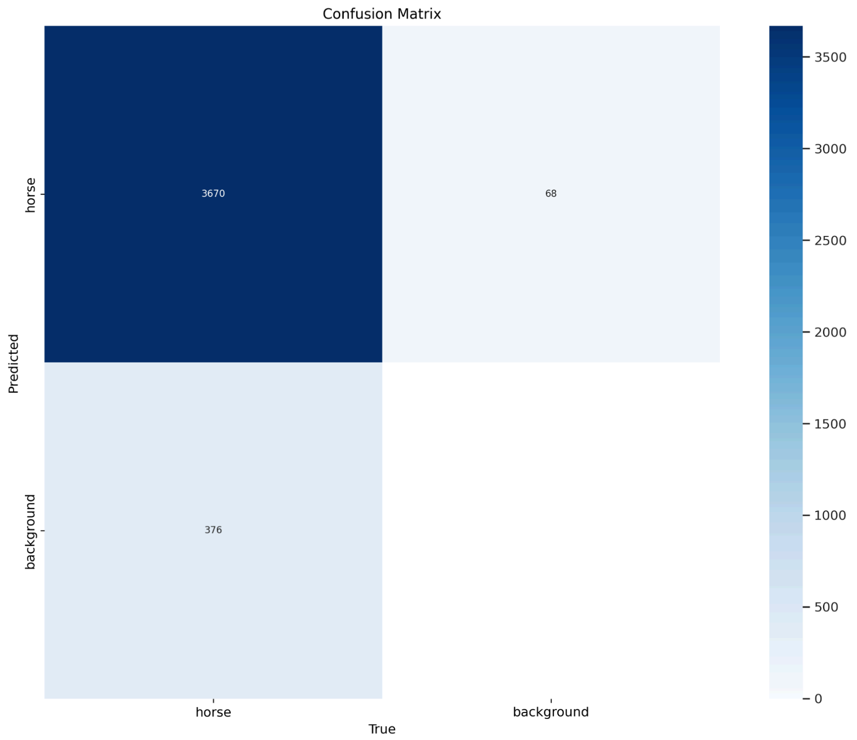
→Overall, the increasing trend followed by stabilization implies that the model ultimately achieves a high level of accuracy in detecting bounding boxes, especially when considering a range of IoU thresholds from 0.5 to 0.95 as it reaches 0.55 and 0.45.

IV- F1-Confidence Curve:



→As we increase the confidence threshold of our YOLOv8 model predictions, the F1-score also increases. It increases to approximately **0.7** for confidence values within the range **[0.05, 0.95**].
This indicates that our predictions become more precise, reducing both false positives and false negatives.

V- Confusion Matrix:



→The model correctly predicted 3670 (out of 4046) instances where horses were present in the image and these instances were indeed horses, and incorrectly predicted 376 instances as background when they were actually horses.
→ It correctly predicted 3978(out of 4046) instances as background when they were indeed background, and incorrectly predicted 68 instances as horses when they were actually background.
→Overall, while the model shows proficiency in detecting horses and background, there is room for improvement to reduce misclassifications, particularly false positives for horses.

VI- Prediction samples:

Bounding Box Estimation:

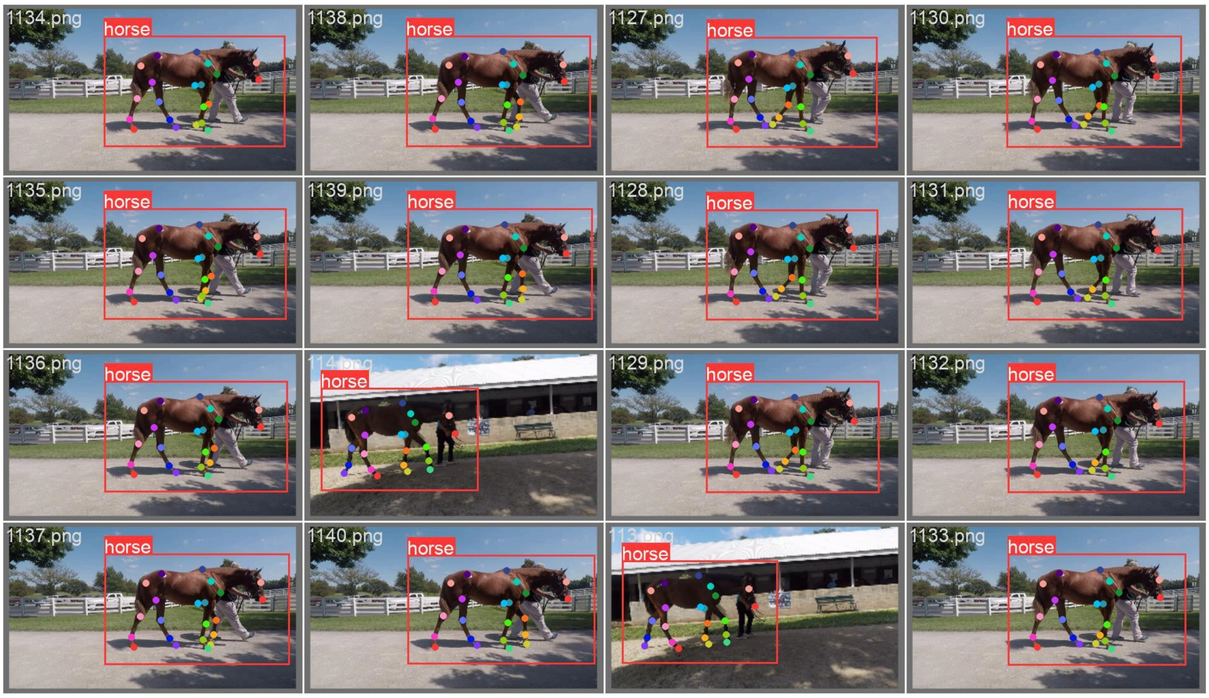
The confidence scores for the bounding boxes are robust, between 0.9 and 1. This indicates that the model is highly accurate in recognizing and localizing the horse in the frames.

Keypoint Estimation:

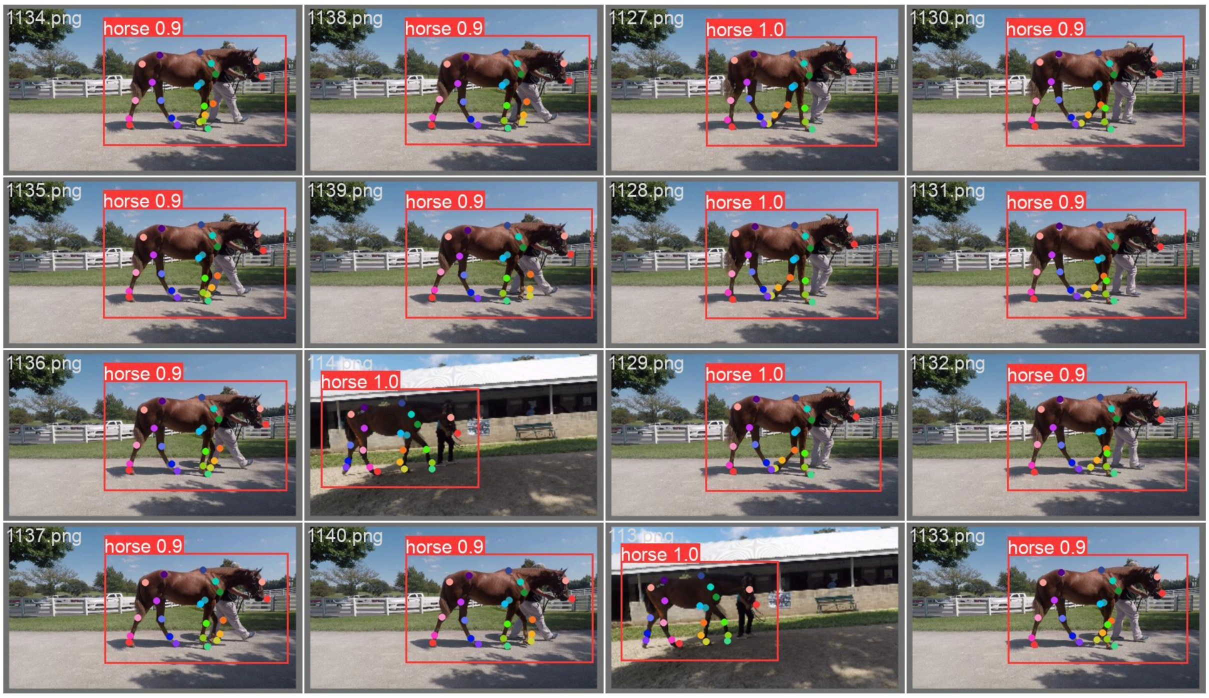
The keypoint detection of dynamic parts like the knees, fetlocks, feet and nose is inaccurate. The errors are more pronounced in the front parts which is likely due to their higher degree of variability during the horse's movement. More stable regions like the torso, hips, neck and eyes are estimated with higher precision

Conclusion :

The model is highly accurate in recognizing the horses and performs better with stable features (keypoints) detection.



- batch labels -



- batch pose estimation-