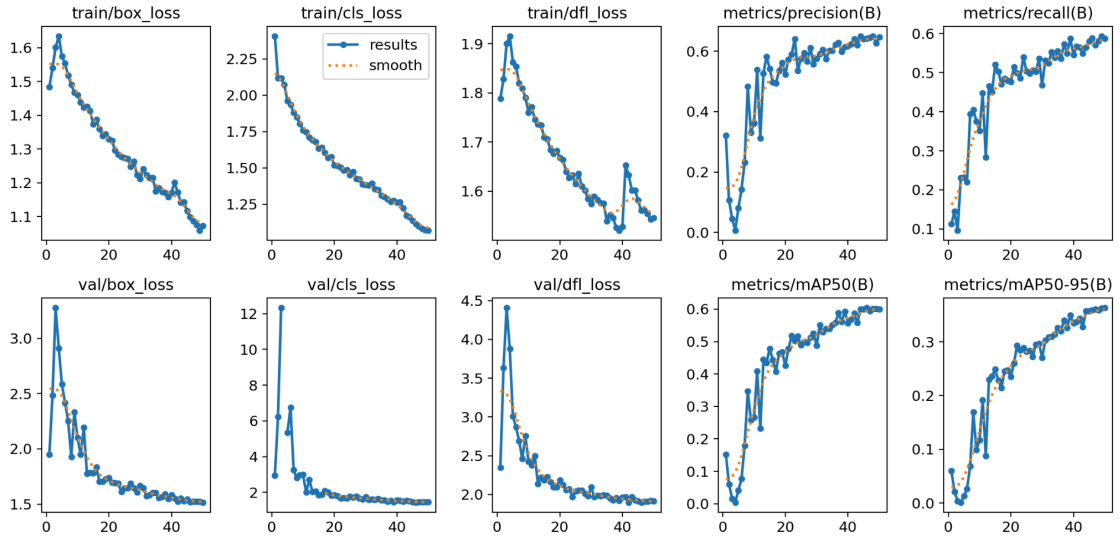
YOLOv8 on Hotdog Object Detection – Model Analysis

Overview

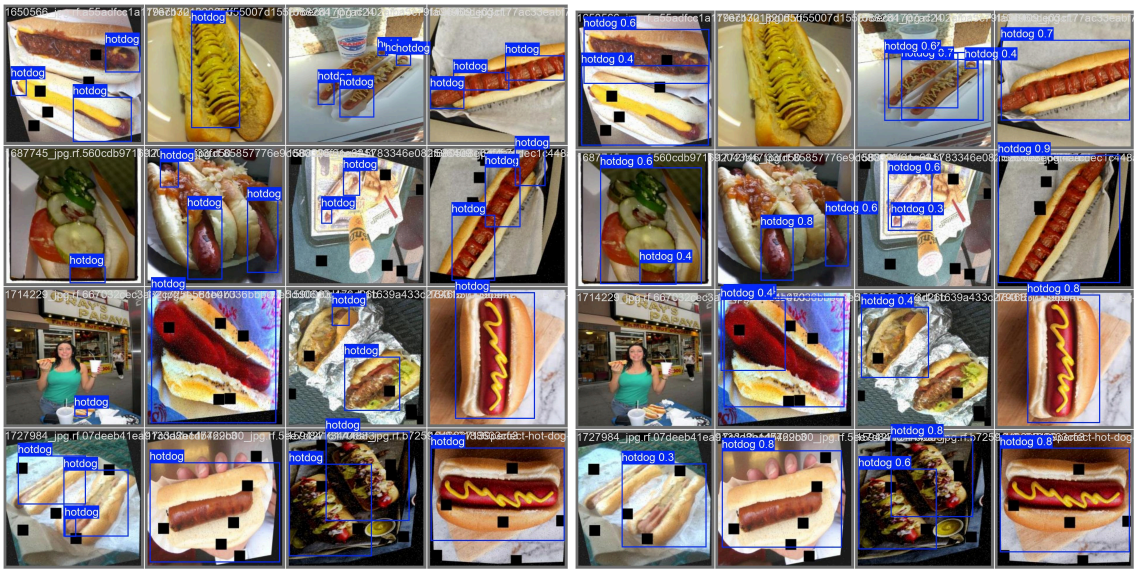
The introduction of the project has been included in README.md and shall not be repeated here. The following data are obtained by training and validating using Google Colab's Nvidia T4 GPU. The metrics are calculated automatically as part of Ultralytics Python API.

It is very important to note that the model has only been trained for 50 epochs, while it has been made clear on the YOLO website that the starting point should be around 300 epochs. This decision was made in order to shorten training time since actual performance does not matter for this task. Most of the metrics discussed below will improve (considerably) after training on the full suggested epoch.

The first graph below shows the classification loss, localization loss, and distribution focus loss of the model generally decrease steadily over time after epoch 15. However, at the end of epoch 50 they still remain greater than 1, which means much more epoch should be required to reduce loss to under 0.01.



Sample Validation Data



The left image above is the first batch of validation data with labels, and the right image is the predictions of the model. This is solely for a high-level visualisation of the performance.

Class-wise Metrics

For the hotdog class, the validation set consists of 714 images and 1251 instances of hotdog objects. The following table records metrics for evaluation performance on object detection.

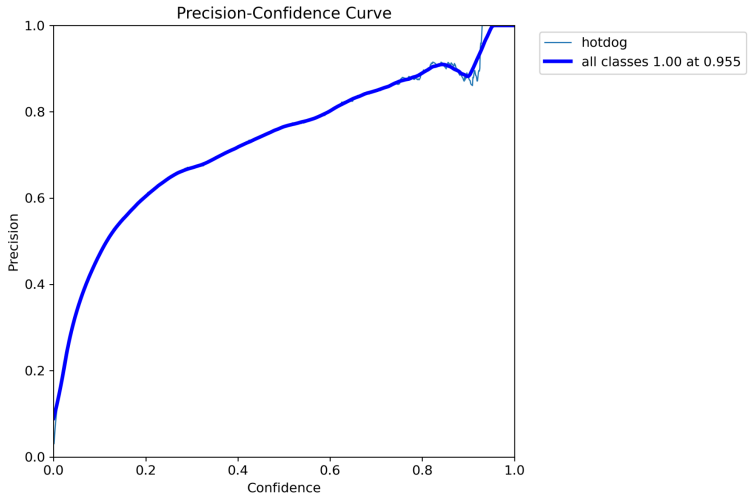
Metric	Value
Precision	0.7658
Recall	0.4548
mAP@0.5	0.6060
mAP@0.5:0.95	0.4347

According to the data, we can make conclusions that:

- Precision (0.76) is good, indicating the model is selective and predicts correctly over 75% of the time when it makes a prediction. However, the tradeoff here is a low recall (0.455), meaning the model misses over half true positives.
- Mean Average Precision (mAP) at Intersection over Union (IoU) threshold 0.5 is 0.6, which shows the model is fairly decent at detecting objects but not exceptional. Only 60% of bounding boxes have at least 50% overlap with the ground truth. Low mAP suggests poor general performance but it should improve with more epochs.
- mAP over IoU 0.5 to 0.95 decreased by 33% compared to mAP@0.5. This highlights the model's difficulty in precisely localising objects, especially at higher IoU thresholds. While the model can detect objects, it struggles to place bounding boxes tightly around the objects

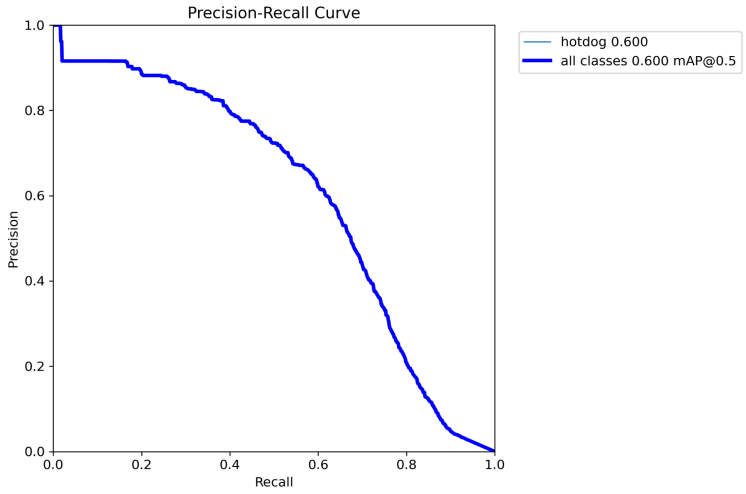
Precision-Confidence Curve

This curve shows how the model's precision varies with different confidence thresholds. Precision is the proportion of true positive detections among all positive detections. It shows the model achieves over 80% precision at confidence 50% or higher. A high precision at all confidence thresholds suggests the model is reliable in making confident detections.



Precision-Recall Curve

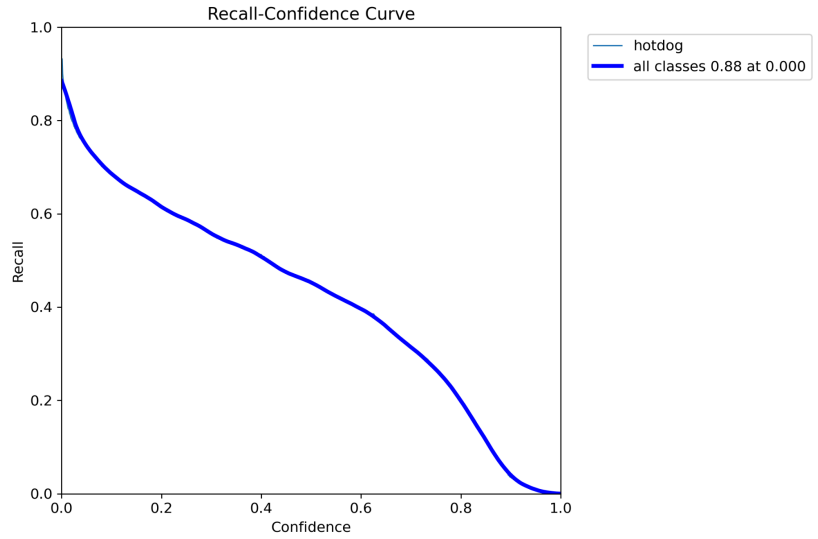
The PR curve illustrates the trade-off between precision and recall across different confidence thresholds. It helps evaluate how well the model balances between detecting all objects (recall) and minimizing false positives (precision). Ideally, the curve should stay high on both axes. A relatively steep drop in precision as recall increases past 0.5 increases suggests the model struggles to detect objects without introducing false positives.



Recall-Confidence Curve

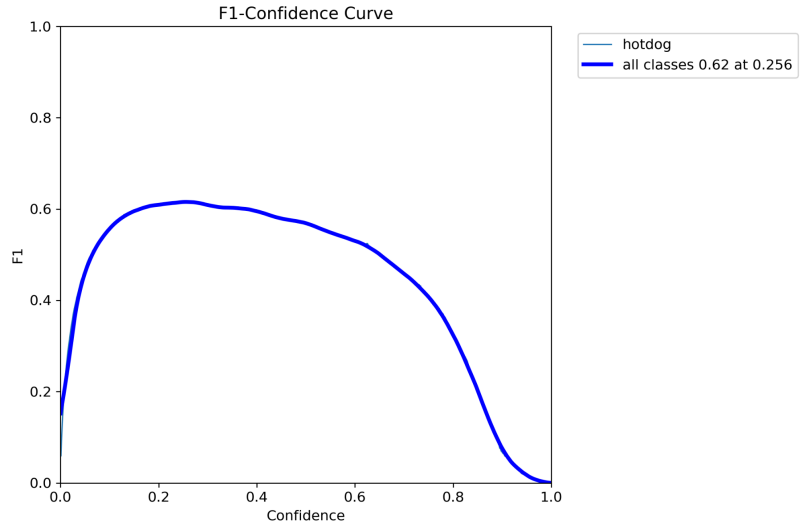
This curve shows how the recall varies with confidence thresholds. Recall is the proportion of true positive detections out of all actual positives. A recall that remains high even at higher confidence thresholds suggests the model is good at detecting most objects. In this case,

since when we raise the confidence threshold to 75% recall drops to only 30%, the model might miss many true positives when confidence is set too high.



F1-Confidence Curve

The F1 score combines precision and recall, providing a single metric to measure model performance. The peak of the F1 curve shows the optimal confidence threshold where both precision and recall are balanced. This is achieved at around 20%, which is far from ideal. A steep decline at both ends indicates that the model's overall performance degrades quickly at too low or too high confidence levels, caused by low precision and low recall respectively.

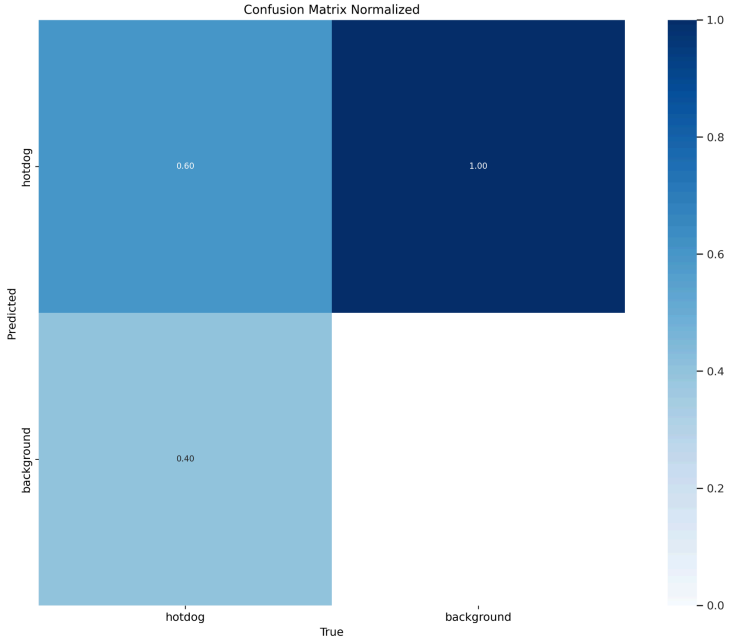


Confusion Matrix

A confusion matrix summarises the performance of a classification model by displaying the counts of true positives, false positives, false negatives, and true negatives. It's a powerful tool to evaluate where the model is making correct predictions and where it is misclassifying. High values in the off-diagonal cells (false positives and false negatives) suggest issues in

distinguishing certain classes. Since the model correctly identified 60% of the hotdogs and missed 40%, the recall for the hotdog class is 0.60. This implies that the model may need improvement in recognizing all instances of hotdogs (i.e., reduce the false negatives).

The cause of the issue with misclassifying background as hotdog is uncertain. Since the model is trained with only one class ("hotdog") and no explicit background class, it may be overly confident that everything it sees belongs to the "hotdog" class. The lack of alternative class in the dataset may cause the incorporation of background regions as hotdogs. It is also possible that the model has overfitted to the features of hotdogs during training, resulting in it being only sensitive to detecting hotdog-like features in non-hotdog areas (false positives).



Inference Time

On the validation set, the recorded speed was 0.3 ms in pre-processing, 4.0ms in inference, and 1.5ms in post-processing per image. The overall process takes 5.8ms and is considered very efficient. Its current performance is within the requirement range for near-instantaneous results for real-time systems. Having low per-image processing times will also allow it to achieve high throughput when processing large numbers of images.