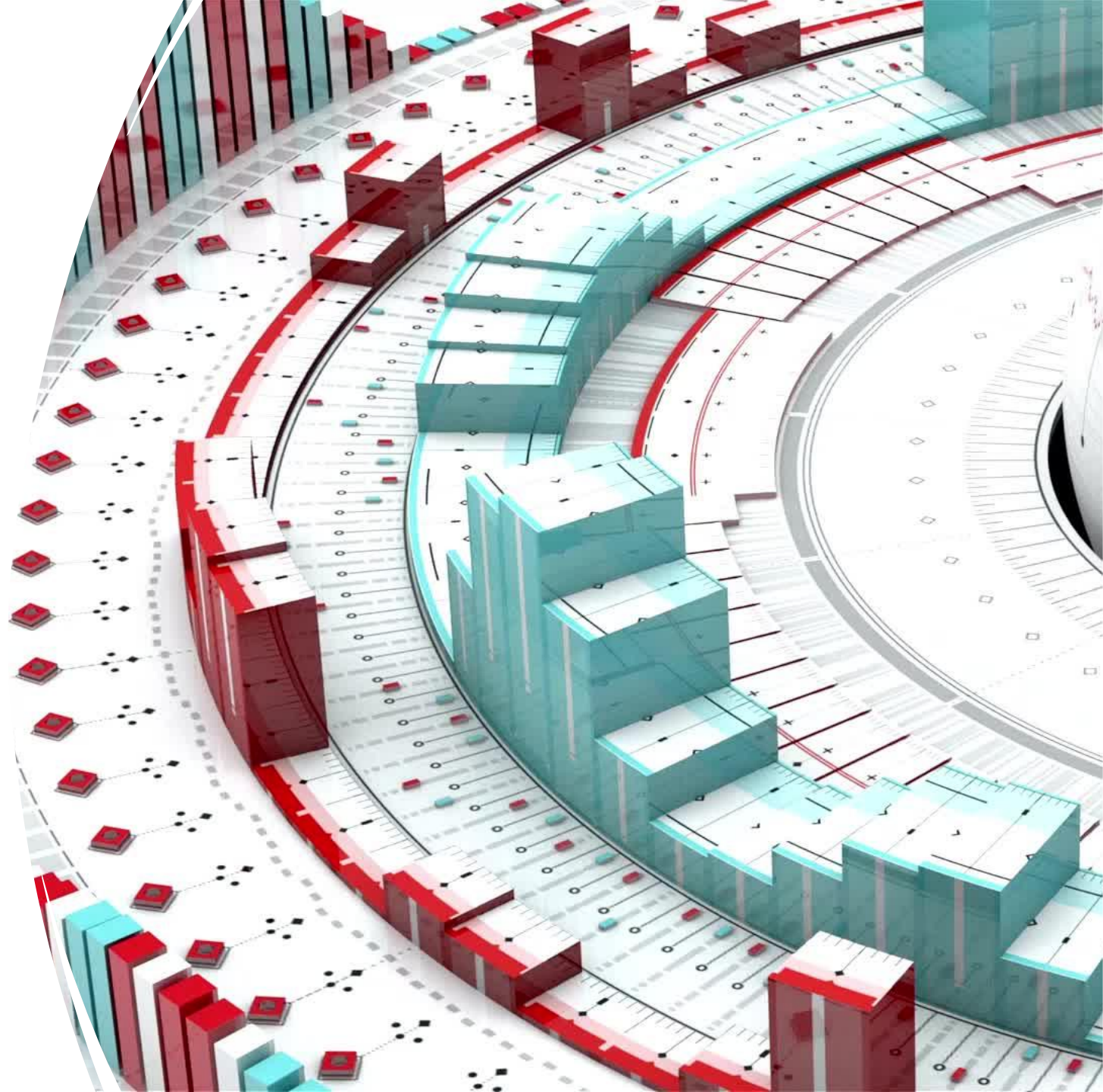


A group of business professionals in an office setting. A woman in a grey blazer is pointing at a tablet held by another person. A man in a dark suit and striped tie is visible on the left. The scene is brightly lit, likely from a window in the background. The text 'Final Intro To CV Course Project' is overlaid in the center.

Final Intro To CV Course Project

The Problem

- Using the Florence-2 model to develop a labeled training dataset for an object detection project from a larger-scale dataset, without the need for human annotations.
- Fine-tuning a pre-trained OD model to detect specific objects that were classified in the dataset creation part of the project.



Dataset Creation – Decisions About df

I decided to filter out images that were below a certain minimum resolution. The goal was to ensure that all training images had enough visual detail to support accurate object detection. Low-resolution images were seen as problematic because they often lack the clarity needed for reliable bounding box predictions, potentially harming the model's learning process.

Dataset Creation – Challenges I've Faced

The inference notebook that you have provided us in the assignment description was lacking some important installations that were crucial for creating a Florence-2 model. This made me need to research a solution, costing me valuable time as a result and making this assignment unnecessarily more complex.

Dataset Creation – TTA

Annotation Verification via TTA:

1. Run Florence-2 on an image → get annotations (e.g., bounding boxes for “person”).
2. TTA Runs:
 1. Apply augmentations like horizontal flip, slight rotation, or scaling.
 2. Run Florence-2 again on the augmented image.
3. Reverse the augmentation transformations (e.g., flip boxes back).
4. Compare Predictions:
 1. Use IoU (Intersection over Union) or object class matching.
 2. If predictions match well across all TTA variants, the annotation is probably correct.
 3. If not, flag for manual inspection or re-annotation.

Dataset Creation – Ensembles

Instead of relying on a single model's prediction, an ensemble collects the outputs of several models and combines them in some way — for example, by:

- Voting (for classification): majority vote across models.
- Averaging (for regression or probabilities): mean of outputs.
- Weighted combination: more reliable models contribute more to the final result.

Why use ensembles?

- They help reduce overfitting (errors specific to a single model).
- They improve generalization by capturing different patterns in the data.
- They mitigate individual model biases or errors, especially in noisy or uncertain tasks like weakly-labeled data or pseudo-labeling.

In our case, if multiple object detection models agree with the Florence-2 annotations, we can be more confident that the annotations are correct.

Model Training – Architecture Choice

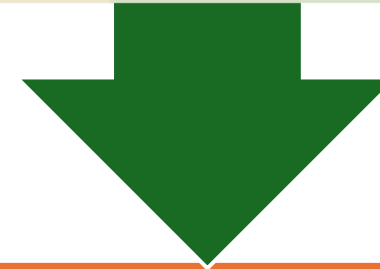
We have chosen Ultralytics's YOLOv8 model because:

It seemed like the easiest model to implement

It is known as an excellent model for real-time object detection

It offers an easy way to perform augmentations and evaluations

It offers a variety of model sizes, which makes it perfect for implementation on virtual GPU's that are limited in use time.



After thoroughly analyzing the performance of the nano model, I decided to leverage the capabilities of YOLOv8-small, complemented by a range of advanced augmentations. This strategic choice aims to enhance accuracy and robustness in our results, allowing for more nuanced insights and improved detection capabilities.

Model Training – Raw Model's Results VS Augmented Model's Results



During the training process, we first trained the YOLOv8-n model on the dataset created using Florence-2 annotations without any custom augmentations. The initial model achieved a solid performance (particularly in detecting the "pet" class) with mAP50 around 0.774 and mAP50-95 at 0.560 overall.



We then proceeded to retrain the model with selected augmentations such as flipping, brightness adjustments, shearing, and mixup to improve generalization.



After augmentation, the model maintained strong performance (mAP50 = 0.778), showing improved robustness and recall for both classes, especially "pet", although mAP50-95 dropped slightly to 0.528. These results suggest that our augmentations introduced more variability and helped the model learn better spatial and visual representations.



Overall, we saw that adding augmentations helped the model become more flexible and better at recognizing people and pets in different situations. However, we also noticed that its performance slightly dropped when it had to make very accurate predictions, which means we need to fine-tune it more if we want better results on stricter accuracy tests.

Model Training – Nano model VS Small model

In the second training experiment, we increased the model size from YOLOv8-n to YOLOv8-s and applied a different set of augmentations.

While the second model had significantly more parameters (11M vs. 3M), its performance gains were modest:

- The precision dropped slightly (from 0.844 to 0.753), but recall improved (from 0.677 to 0.711).
- The mAP50 stayed nearly the same (0.778 vs. 0.774), and mAP50-95 improved slightly (from 0.528 to 0.548).

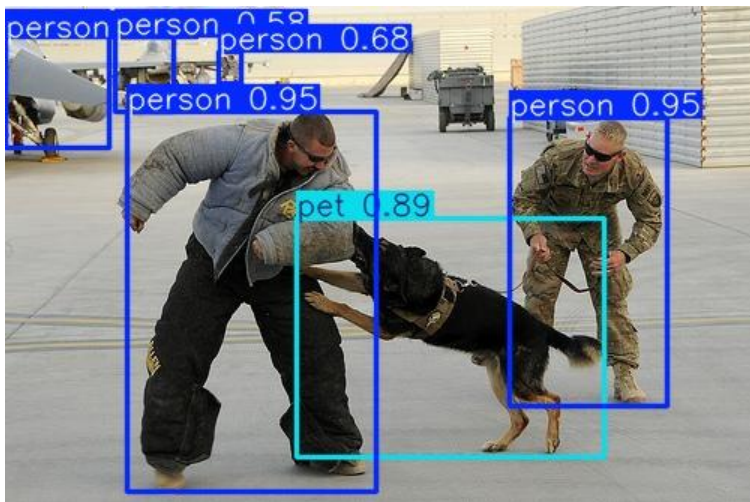
This suggests that the larger model with alternate augmentations captured finer details (as seen in the improved mAP50-95), but at the cost of lower precision, likely due to increased false positives.

Overall, both models performed well, with different strengths.

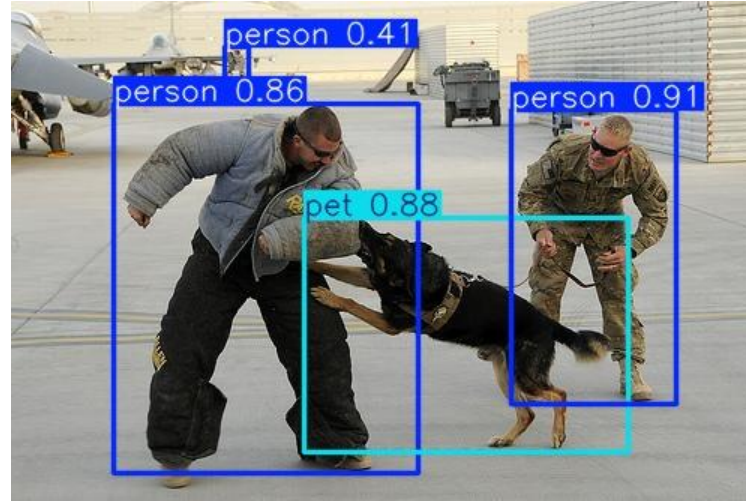
Exemplary inference image and mAP50 of Models

mAP50	All	Person	Pet
Raw Model	0.774	0.776	0.773
Nano Augmented	0.778	0.774	0.783
Small Augmented	0.774	0.793	0.756

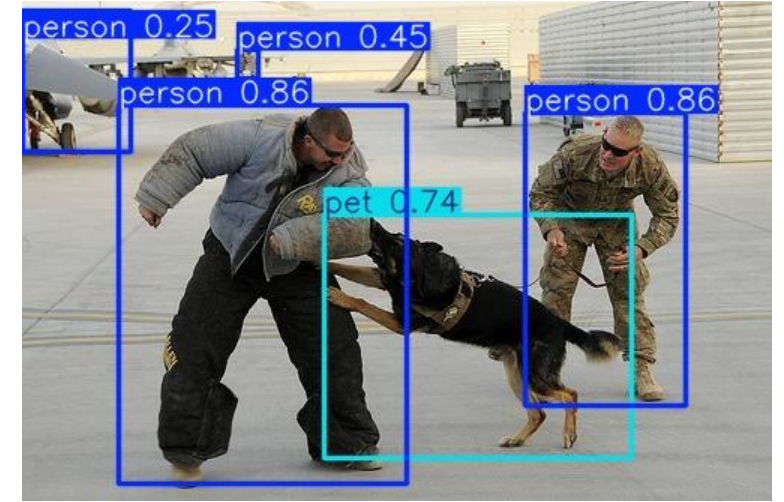
Model trained without augmentations:



Model trained with augmentations:




Model trained with more parameters (YOLOv8-





Future Work



If we had more time, I would have tested a larger YOLOv8 model, such as the medium or large versions. Additionally, I would have attempted to implement ensemble methods to enhance the accuracy of the labels and to better understand their impact on the outcome.

Reflection

My conclusion:

- Annotations created without human intervention can be reliable.

What I've learned:

- This project has enhanced my understanding of the tools we've learned in this course and how they can be applied to future endeavours. Particularly, it will be beneficial for my final degree project, which shares several similarities with this assignment.

What would I have done differently:

- I would have likely tried to give this project a practical context by connecting or creating software that relies on these capabilities.