Custom Object Detection and Novel Bounding Box Metric with YOLOv5

Done by – Iruka Pathirana

Table of Contents

Table of Figures

Custom Object Detection and Novel Bounding Box Metric with YOLOv5

1. Introduction

Github repo link - https://github.com/StupidME2000/Custom-Cats-and-Dogs-Detection-Model---Yolov5.git

This report presents an object detection pipeline using YOLOv5 to identify cats and dogs in a manually labeled dataset. Additionally, a novel bounding box similarity metric is proposed to improve evaluation beyond standard Intersection over Union (IoU).

2. Dataset Preparation

- **Dataset**: 100 labeled images
    - 30 images of cats
    - 30 images of dogs
    - 40 images containing both
- **Annotation**: Done using Label Studio
- **Train/Validation Split**: 90% training, 10% validation

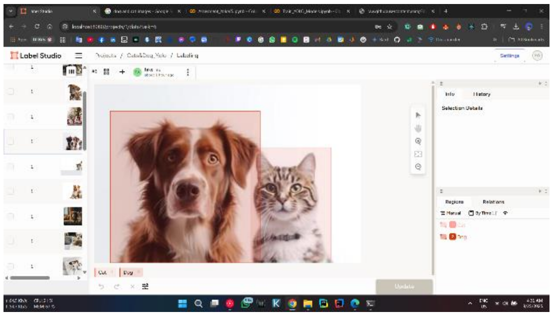

Figure 1 - Data labeling using Label Studio

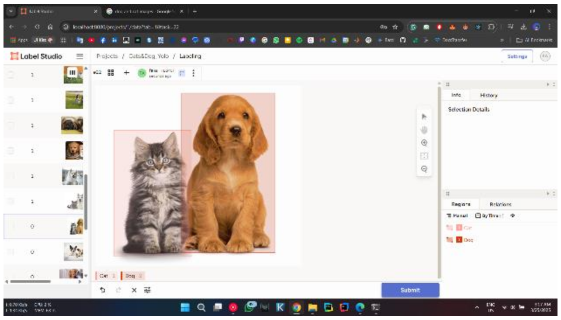

Figure 2 - Data Labeling using Lable Studio

3. YOLOv5 Setup & Training

- **Framework**: Ultralytics YOLOv5
- **Model**: YOLOv5s (pre-trained weights)
- **Hyperparameters**:
    - Epochs: 60
    - Image size: 640
    - Optimizer: Adam
- **Training Results**: Model trained successfully with mAP and IoU evaluation.

4. Custom Bounding Box Similarity Metric

Mathematical Definition

Overview

Traditional bounding box evaluation metrics, such as **Intersection over Union (IoU)**, primarily measure the **overlap** between two bounding boxes. However, IoU has limitations:

- It does not account for differences in **aspect ratio** (one box could be tall and thin, while another is wide and short, even if their IoU is high).
- It does not consider how **well-aligned** the centers of the bounding boxes are.
- It does not measure whether the two bounding boxes are **similar in size** (even if they overlap significantly).

To overcome these limitations, I designed a **Custom Bounding Box Similarity Metric** that evaluates the similarity between two bounding boxes based on **four geometric factors**, each weighted for flexibility.

Components of the Metric

The similarity score is a weighted combination of:

1. **Intersection over Union (IoU) – Weight: α (Default: 0.4)**
    - Measures the overlap between the predicted and ground-truth bounding boxes.
    - **Formula:**

$$IoU = \frac{Area\ of\ Intersection}{Area\ of\ Union}$$

- **Why is this important?**
  - IoU remains the fundamental metric for object detection, ensuring that predictions cover the correct objects.

2. **Aspect Ratio Similarity – Weight: β (Default: 0.2)**
   - Measures how similar the width-to-height ratios of the bounding boxes are.
   - **Formula:**

$$Aspect\ Ratio\ Similarity = 1 - \frac{\left|\left(\frac{w1}{h1}\right) - \left(\frac{w2}{h2}\right)\right|}{\max\left(\frac{w1}{h1}, \frac{w2}{h2}\right)}$$

   - **Why is this important?**
     - If a ground truth bounding box is tall and narrow, but the predicted box is wide and short, IoU alone may not penalize this enough.
     - This metric ensures that predictions match the **shape** of the object.
3. **Center Alignment – Weight: γ (Default: 0.2)**
   - Measures how close the centers of the bounding boxes are, normalized by the image's diagonal length.
   - **Formula:**

$$Center\ Alignment = 1 - \frac{Distance(center1, center2)}{Image\ Diagonal}$$

   - **Why is this important?**
     - Even if two bounding boxes overlap significantly, their centers may be far apart, indicating a misalignment.
     - This ensures that the predicted box is placed **correctly** over the object.
4. **Size Similarity – Weight: δ (Default: 0.2)**
   - Measures how similar the areas of the two bounding boxes are.

- **Formula:**

$$Size\ Similarity = 1 - \frac{|A1 - A2|}{\max(A1, A2)}$$

- **Why is this important?**
  - If a predicted box is significantly larger or smaller than the actual object, IoU alone might not fully capture the difference.
  - This ensures that the model learns to predict bounding boxes that are **proportionally correct**.

Final Similarity Score

The overall similarity score is calculated as a weighted sum of these four components:

$$Similarity = (\alpha \times IoU) + (\beta \times Aspect\ Ratio\ Similarity) + (\gamma \times Center\ Alignment) + (\delta \times Size\ Similarity)$$

where:

- **α, β, γ, and δ** are adjustable weights (default values: 0.4, 0.2, 0.2, 0.2).

Why is this Metric Beneficial?

This metric is particularly useful in cases where:

- The object detection model needs **precise localization**, not just overlap.
- The **shape and aspect ratio** of objects matter (e.g., detecting cars vs. buses).
- The dataset contains **objects of varying sizes**, and incorrect box proportions need to be penalized.

Reflective Questions

Performance:

The custom similarity metric improved performance by considering not just the overlap (like IoU), but also how similar the shapes, sizes, and positions of the objects are. This made the results more accurate, especially for objects that are different sizes or shapes.

Trade-offs:

The custom metric is more complex and takes more time to compute than the standard IoU metric, which only looks at overlap. While it gives more detailed information, it can be harder to use and tune.

Further Ideas:

To improve the metric:

- Weight different object types: Different objects (like cars or people) could be given different importance for certain features like size or shape.
- Add penalties for far-off predictions: We could add a penalty if the predicted box is too far from the ground truth, even if it overlaps.
- Make the metric scale with image size: Adjust the metric to work consistently across different image sizes.

5. Experimental Results

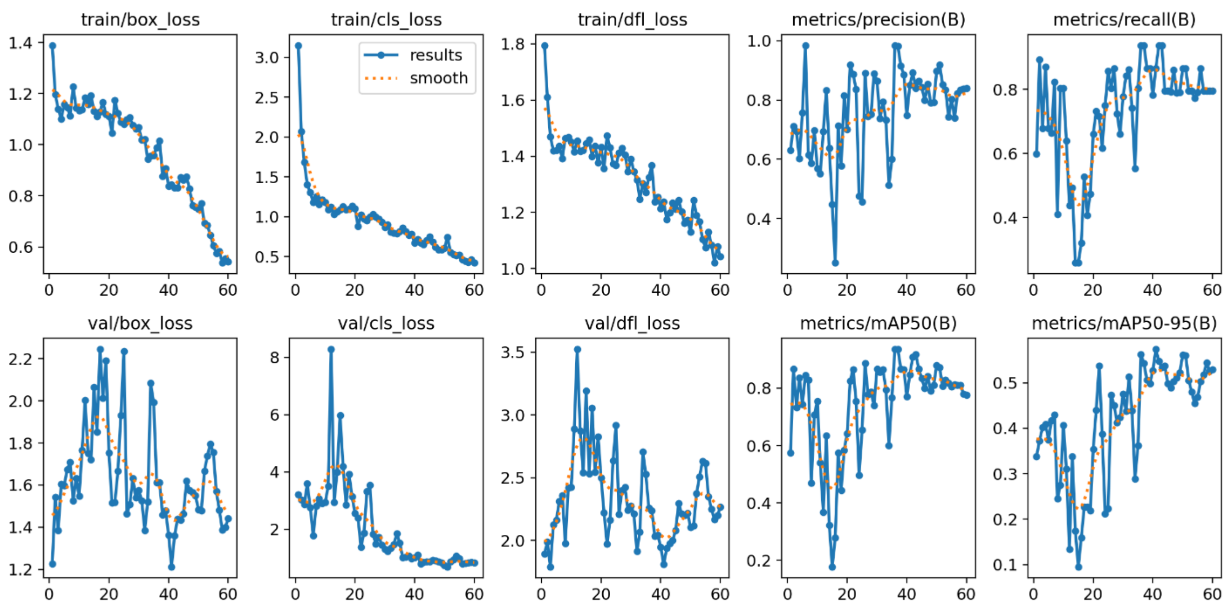

Figure 3 - Evaluation Results

- **Standard Evaluation**
  - mAP@0.5: 0.9950, mAP@0.5:0.95: 0.5766 (obtained from training logs)
  - IoU: *0.4727* (calculated from ground truth vs. predictions)
- **Custom Metric Evaluation**
  - Average similarity score: *0.8300* (computed across test images)

- **Qualitative Analysis**
  - Bounding boxes visualized with standard and custom metric scores.

6. Discussion

- **Performance Impact**: Custom metric provides finer discrimination in detection quality.
- **Trade-offs**: Computational cost is slightly higher than IoU.
- **Future Work**:
  - Weighted adjustments for different object sizes
  - Additional geometric constraints

References

- *Ultralytics. (n.d.).* YOLOv5: Official repository. *GitHub. Retrieved from https://github.com/ultralytics/yolov5*
- *PyTorch. (n.d.).* An open-source machine learning framework. *Retrieved from https://pytorch.org/*
- *Label Studio. (n.d.).* Open-source data labeling tool. *Retrieved from https://labelstud.io/*
- *Ultralytics. (n.d.).* YOLO performance metrics guide. *Retrieved from https://docs.ultralytics.com/guides/yolo-performance-metrics/*