**Higher Recall = Detect more objects**

**Higher Precision = Correctly detect objects**

A graph of a curve

AI-generated content may be incorrect.**🔹 What the chart shows:**

* **X-axis (Confidence):** Threshold for deciding whether a detection is valid.
* **Y-axis (F1 Score):** Harmonic mean of precision and recall (best when both are high).

**🔵 Blue Line:**

* Represents the **average F1 score across all classes**.
* Peaks at **0.73 F1 score** when the confidence threshold is around **0.38**.
* This suggests your model performs best on average when it only reports predictions with **≥38% confidence**.

**⚪ Gray Lines:**

* Represent **F1 performance per individual class** (e.g., stop sign, speed sign, yield).
* Show how some classes perform better than others across the confidence range.
* Variability is expected — especially if some signs are underrepresented or harder to detect (blurred, angled, etc).

**✅ What this means for your project:**

* **Recommended Threshold:** Based on this curve, you could set the model confidence to **0.4** in production to balance precision and recall.
* **Model Quality:** An average F1 score of **0.73** is strong for a real-time multi-class detection task.
* **Next Steps:** You could investigate the low-performing classes (those with flatter gray lines) and improve them with more samples or augmentation.

A group of graphs showing the size of a number of objects

AI-generated content may be incorrect.  
This image shows the **training and validation performance** of your YOLO model over 50 epochs. Each subplot tells you something different about how your model is learning. Here's a breakdown:

**🔧 Loss Curves (Top row: training | Bottom row: validation)**

1. **box\_loss**: Measures how well the model predicts bounding box positions.
   * 🔻 Decreasing trend → The model is improving at localizing objects.
2. **cls\_loss**: Classification loss — how well it classifies the correct label.
   * 🔻 Strong drop early → Model quickly learns to recognize classes.
3. **dfl\_loss** (Distribution Focal Loss): A YOLOv8-specific loss for bounding box regression precision.
   * 📉 Gradually decreasing → Finer improvements in box predictions.

🟢 **Interpretation for all loss graphs**:  
Loss is going down steadily → ✅ Training is stable and improving. No overfitting signs (losses on training and validation decrease similarly).

**📊 Metric Curves (Performance on validation set)**

1. **precision(B)**: Correct predictions / all predicted (how accurate the model is when it makes a prediction).
   * 🔼 Increasing to ~0.75 → Fewer false positives.
2. **recall(B)**: Correct predictions / all actual objects (how many real objects were detected).
   * 🔼 Rising to ~0.67 → More real signs are being detected.
3. **mAP@0.5 (B)**: Mean Average Precision at 50% IoU threshold.
   * 📈 Rises to ~0.87 → Very good detection performance.
4. **mAP@0.5:0.95 (B)**: Stricter metric (average over IoUs from 0.5 to 0.95).
   * 🔼 Goes up to ~0.69 → Model performs consistently even with tight bounding box matching.

**📌 Summary for Your Presentation Slide**

**Title: Training Performance Summary**

**Key Points:**

* The model's training and validation losses steadily decreased, showing effective learning.
* Precision and recall improved across epochs — the model became both more accurate and more complete in detecting traffic signs.
* High **mAP@0.5 (~87%)** and solid **mAP@0.5:0.95 (~69%)** indicate strong object detection performance, even under strict evaluation.
* No signs of overfitting — validation metrics closely follow training performance.

Want me to turn this into a formatted slide content block too?

**What is** [**mAP@0.5**](mailto:mAP@0.5) **and** [**mAP@0.5:0.95**](mailto:mAP@0.5:0.95)**?**

**What is mAP?**

* **mAP** = Mean Average Precision
* It combines **precision and recall** into one number that tells you **how good your model is at detecting and correctly classifying objects**.

**🧠 How it works:**

1. For each object class (e.g., "Stop", "Speed Limit 50", etc.):
   * The model makes predictions with a certain **confidence score**.
   * It compares those predictions with the actual (ground truth) boxes.
   * For various thresholds of **IoU (Intersection over Union)**, it checks:
     + Did the model detect the object?
     + Was the box close enough (IoU > threshold)?
     + Was the label correct?
2. It then computes a **Precision-Recall (PR) curve** for each class and calculates the **area under this curve**, which gives the **Average Precision (AP)** for that class.
3. Finally, it averages the APs of all classes to get **mean Average Precision (mAP)**.

**📏 Two Common Types of mAP in YOLO:**

| **Metric** | **Meaning** |
| --- | --- |
| **mAP@0.5** | Average Precision at IoU threshold of **0.5** (50% overlap is acceptable). Easier. |
| **mAP@0.5:0.95** | Average Precision averaged over **IoUs from 0.5 to 0.95** in steps of 0.05. Stricter and more realistic. |

**📊 What Your Values Mean:**

* **mAP@0.5 ≈ 0.87 (87%)**  
  → Your model detects and correctly classifies signs **very well** when only moderate overlap (50%) is required.
* **mAP@0.5:0.95 ≈ 0.69 (69%)**  
  → Even with tighter requirements on box accuracy, your model performs **strongly**, showing **robust localization**.

**📦 What is IoU (Intersection over Union)?**

**IoU** is a metric used to measure **how well your model’s predicted bounding box matches the actual object** (ground truth box) in an image.

Think of it as a comparison of two rectangles:

* One is the **predicted bounding box** from your model.
* The other is the **ground truth box**, i.e., where the object *actually* is.

**🔲 Formula for IoU:**

IoU=Area of OverlapArea of Union\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}IoU=Area of UnionArea of Overlap​

**🧠 Explanation with Example:**

Imagine:

* You have a **Stop sign** in an image.
* The model predicts a box that *almost* covers it, but not perfectly.

1. **Intersection**: The overlapping region of predicted & actual boxes.
2. **Union**: Total area covered by both boxes combined (without double-counting the intersection).

| **If the overlap is...** | **IoU value** | **What it means** |
| --- | --- | --- |
| Exact match | 1.0 | Perfect prediction |
| 50% overlap | 0.5 | Model caught the object reasonably well |
| Very little overlap | < 0.3 | Poor prediction |
| No overlap | 0.0 | Model totally missed the object |

**📐 IoU in mAP@0.5**

* The **0.5** here means:  
  → A prediction is considered **correct (True Positive)** *only if* its IoU with the actual object is **≥ 0.5**.

So when we say:

"**Average Precision at IoU threshold of 0.5**"

It means:

* Only predictions with at least **50% overlap** with the ground truth box are counted toward precision and recall.
* It's like saying, *"If my predicted box is close enough (≥50% correct), I’ll accept it as a good hit."*