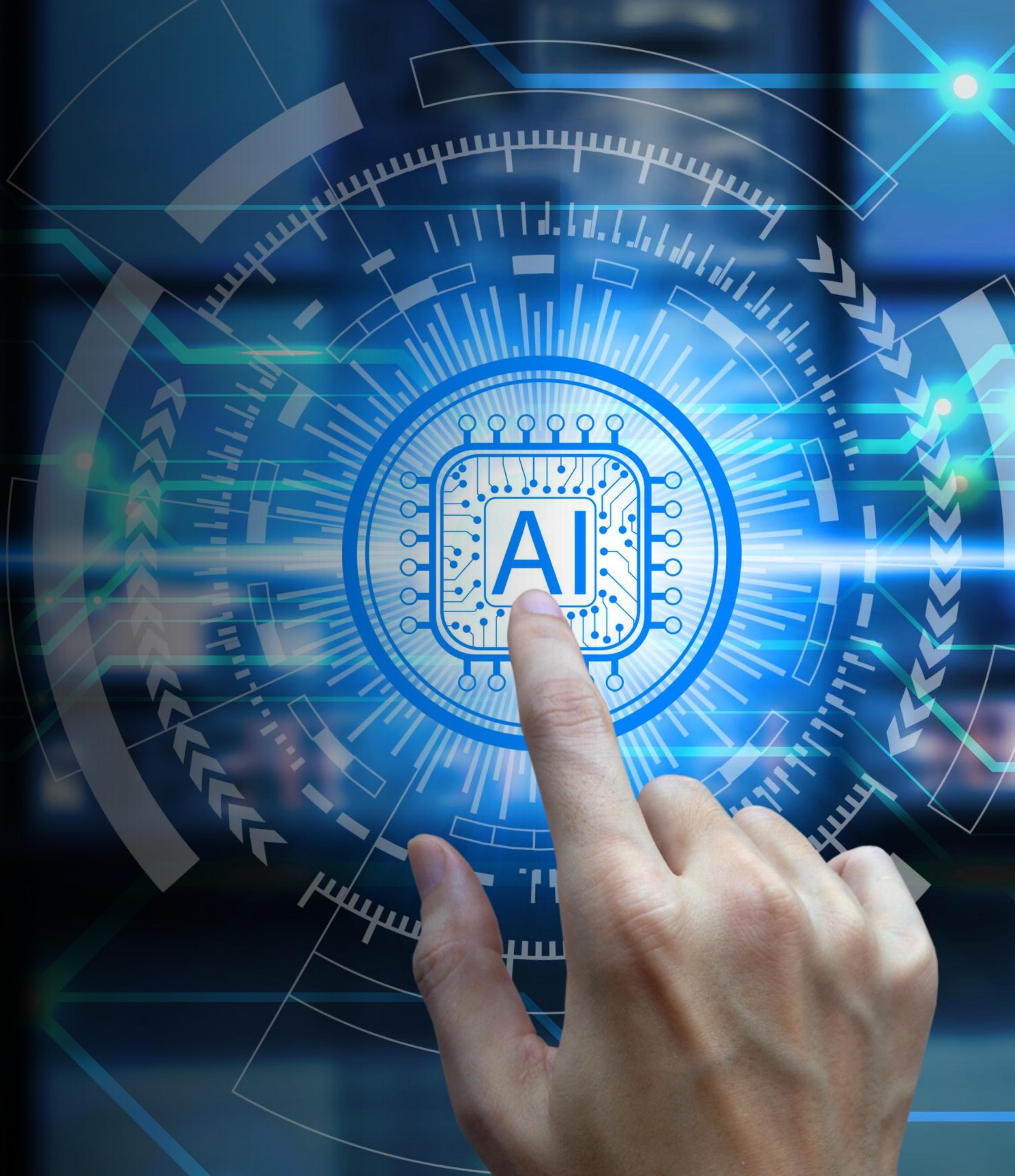
Introduction to Neural Networks SEN4107

DETECTING BRANDS ICONS/ LOGOS ON APPAREL

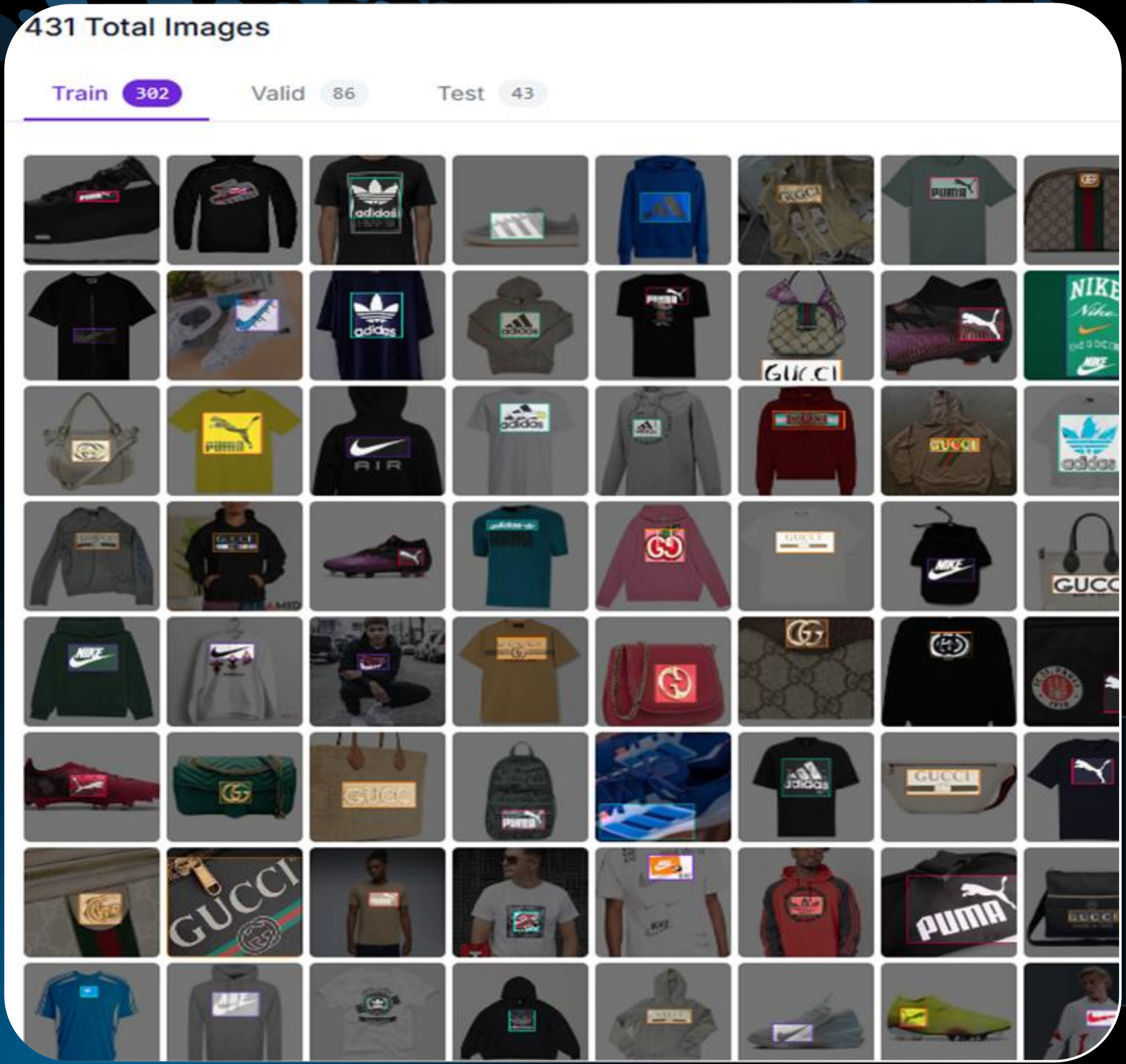


Presented By

Names & IDs:

- Youssef Oueslati – 2367751
- Osama Saleh Khayri AlTamimi – 2019176

INTRODUCTION



Logo detection on clothing is an important computer vision task used in areas such as retail analytics, brand monitoring, and copyright protection. The goal is to automatically detect and classify brand logos appearing on apparel, even when they are small, rotated, or partially hidden.

Dataset

The dataset used in this project was created and managed using Roboflow. It contains images of clothing annotated with four brand logo categories:

Adidas, Nike, Gucci, and Puma. Each logo instance is labeled with a bounding box. The dataset was exported in YOLO format and split into training, validation, and test sets, with data augmentation applied through Roboflow to improve model robustness.

EVALUATION METRICS:

$$\begin{aligned} precision &= \frac{TP}{TP+FP} \\ recall &= \frac{TP}{TP+FN} \\ F1 &= \frac{2 \times precision \times recall}{precision + recall} \\ accuracy &= \frac{TP+TN}{TP+FN+TN+FP} \\ specificity &= \frac{TN}{TN+FP} \end{aligned}$$

Model performance is evaluated using standard object detection metrics:

- Precision — proportion of correct detections
- Recall — proportion of detected true logos
- F1-score — harmonic mean of precision and recall
- mAP@0.5 and mAP@0.5–0.95 — mean Average Precision at fixed and multiple IoU thresholds

Confidence-based curves (Precision–Recall, F1–Confidence) and confusion matrices are also reported to analyze class-wise behavior.

Objective

The goal of this project is to train and evaluate two YOLO-based models — a lightweight baseline and a higher-capacity model — and compare their detection performance across logo classes.

2. RELATED WORK :

Early logo detection approaches relied on handcrafted features such as SIFT and HOG combined with SVM classifiers. However, these methods performed poorly when logos were small, rotated, or distorted. The introduction of deep learning led to CNN-based detectors such as Faster R-CNN, SSD, and YOLO, which greatly improved robustness. In particular, single-stage YOLO models became widely used due to their real-time performance and high accuracy. Recent versions such as YOLOv8 further enhance feature extraction and training efficiency, making them well suited for logo detection. These developments motivate the use of YOLO-based models in this work.

3. MODELS :

3.1 Baseline Repository

The models were implemented using the official Ultralytics YOLO library:
https://github.com/ultralytics/ultralytics
This repository provides pretrained YOLOv8 models and APIs for training, evaluation, and inference.

3.2 Model-1: YOLOv8 (Baseline)

Model-1 loads the pretrained YOLOv8-Nano checkpoint (yolov8s.pt). Training was performed using the script train_yolo_model1.py with:

- Epochs: 60
- Image size: 640×640
- Batch size: 8
- Optimizer: default YOLO optimizer
- Device: CPU
- Data: data/logos/data.yaml
- Augmentation: enabled

3.3 Model-2: YOLOv8-Medium

Model-2 loads the pretrained YOLOv8-Medium checkpoint (yolov8m.pt) and is trained using train_yolo_model2.py with:

- Epochs: 70
- Image size: 768×768
- Batch size: 8
- Device: CPU

This model has greater learning capacity and deeper layers.

3.4 Loss Function

YOLOv8 jointly optimizes:

- Bounding Box Regression Loss
- Classification Loss
- Distribution Focal Loss

Training curves for these losses are reported in the Experiments section.

3.5 Evaluation Pipeline

Model testing was automated in test_model.py, which computes:

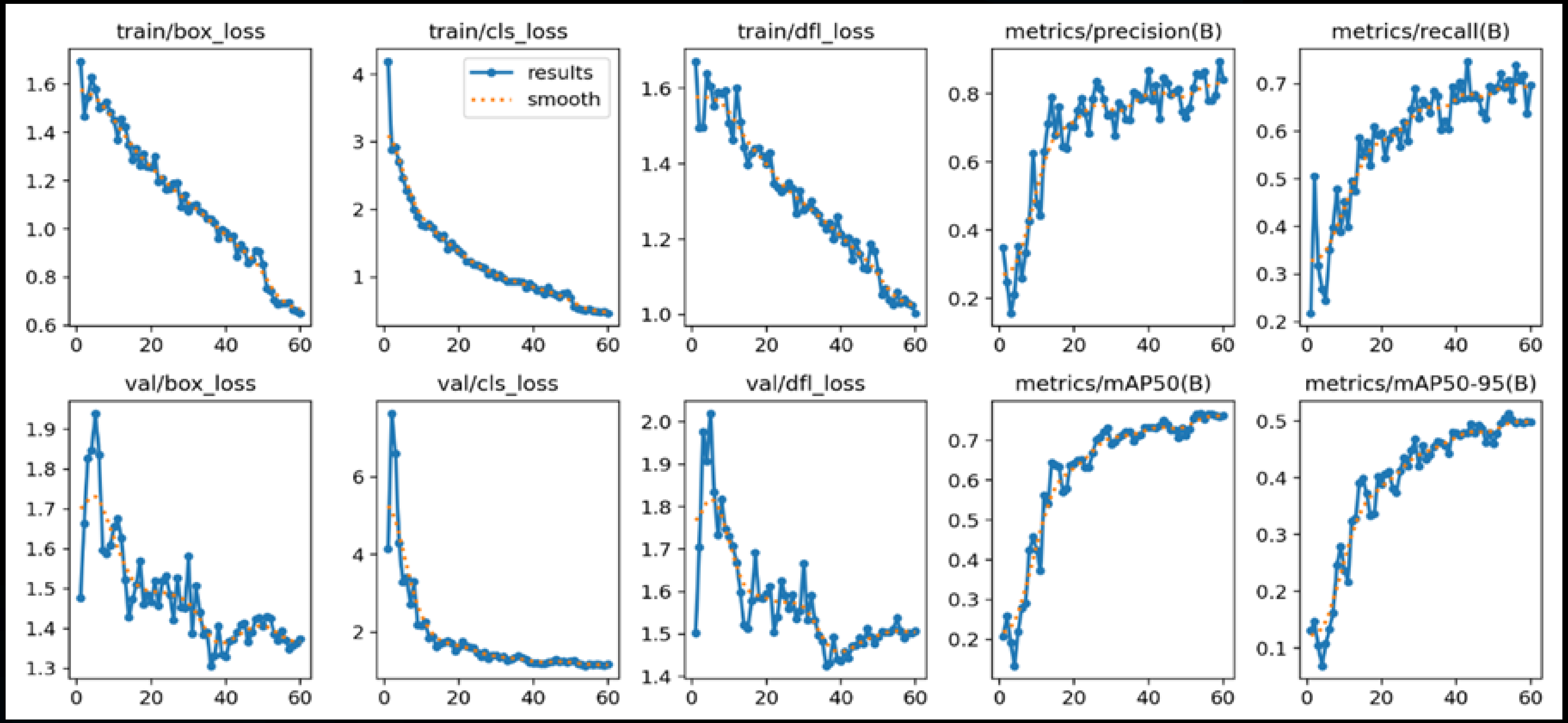
- Precision
- Recall
- mAP50
- mAP50-95

Predictions were generated using predict_logos.py, saving bounding-box overlays.

4. EXPERIMENTS:

4.1 Model-1 : YOLOv8 Training Results:

Figure 1 – Training and Validation Curves (Model-1)



The training and validation losses (box, cls, and DFL) consistently decrease over epochs, showing good convergence without overfitting. At the same time, the evaluation metrics steadily improve: precision rises to ~0.85, recall to ~0.70, mAP50 reaches ~0.74, and mAP50-95 reaches ~0.50. This indicates that the model learns effectively and achieves strong overall detection performance.

The Precision–Recall curves show that Puma and Nike achieve the highest AP values, while Gucci performs worst. The mean AP@0.5 is approximately 0.761, confirming good overall detection accuracy for Model-1.

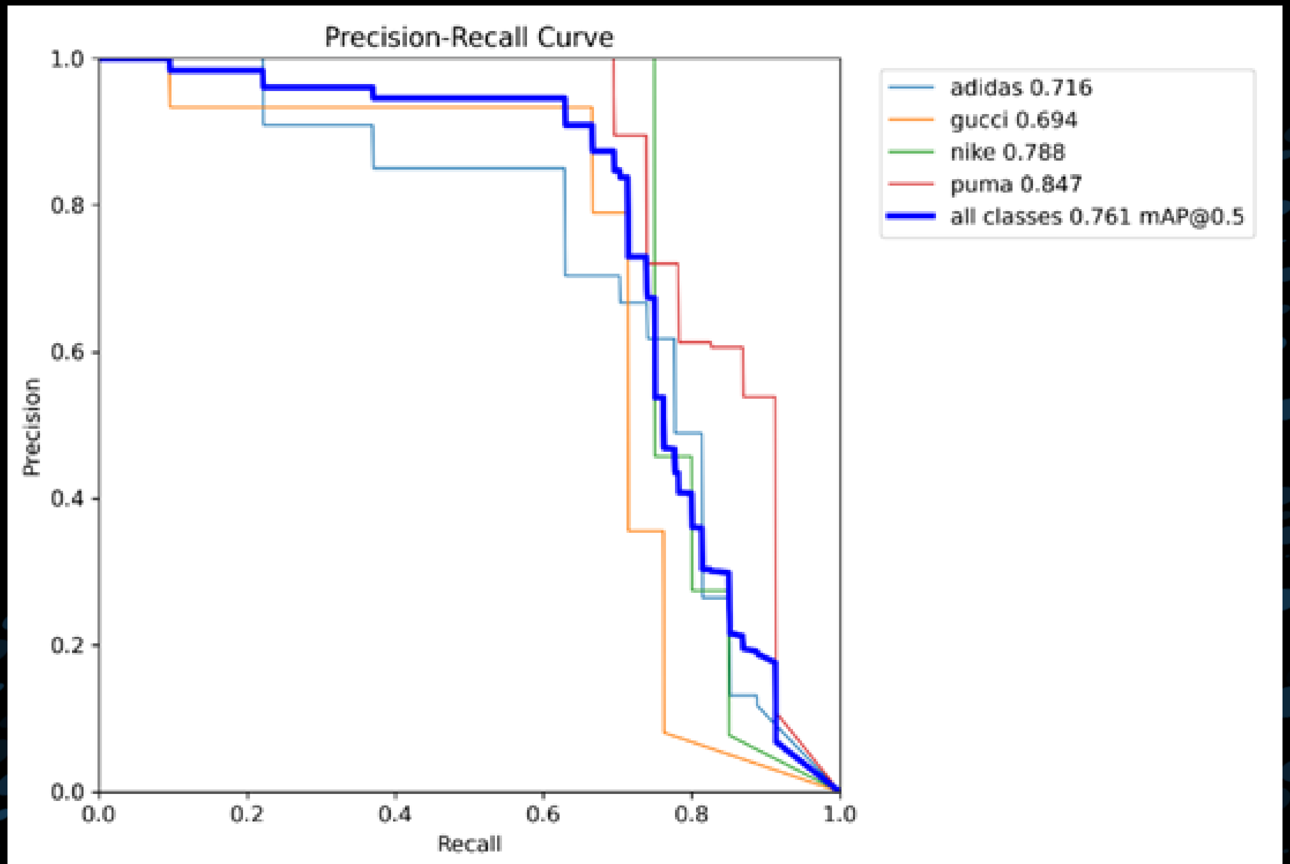
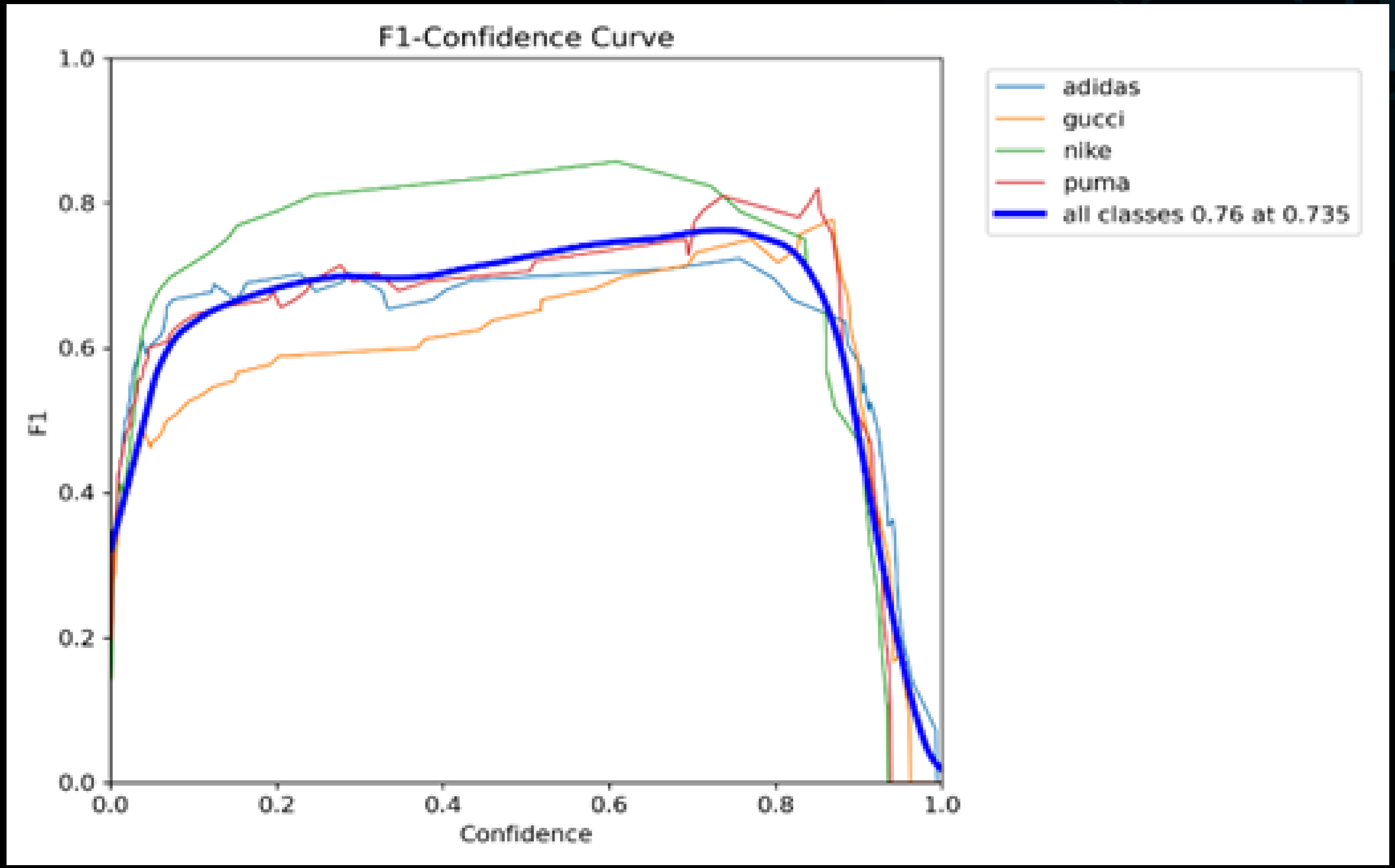


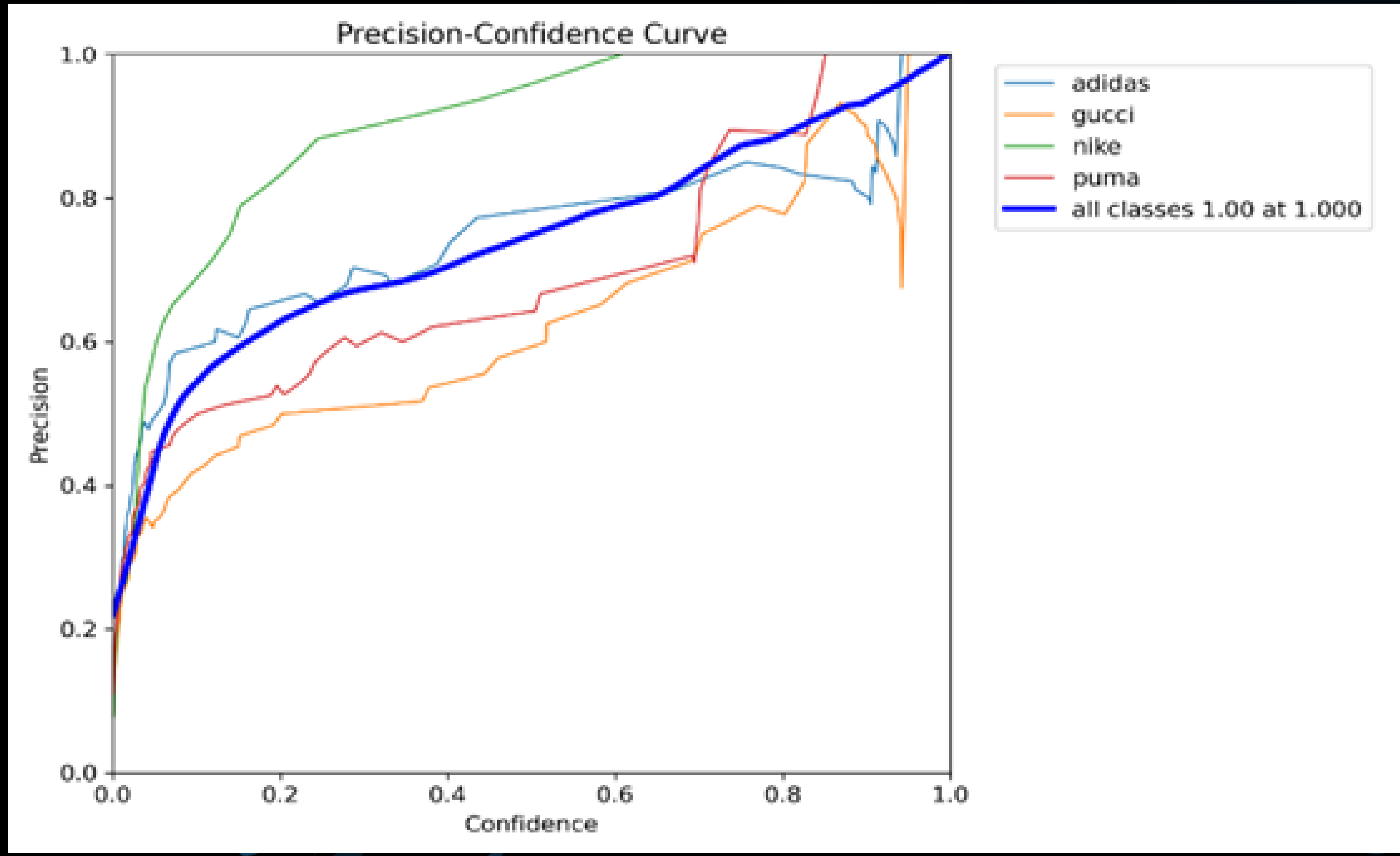
Figure 2 – Precision–Recall Curve (Model-1)

Figure 3 – F1–Confidence Curve (Model-1)



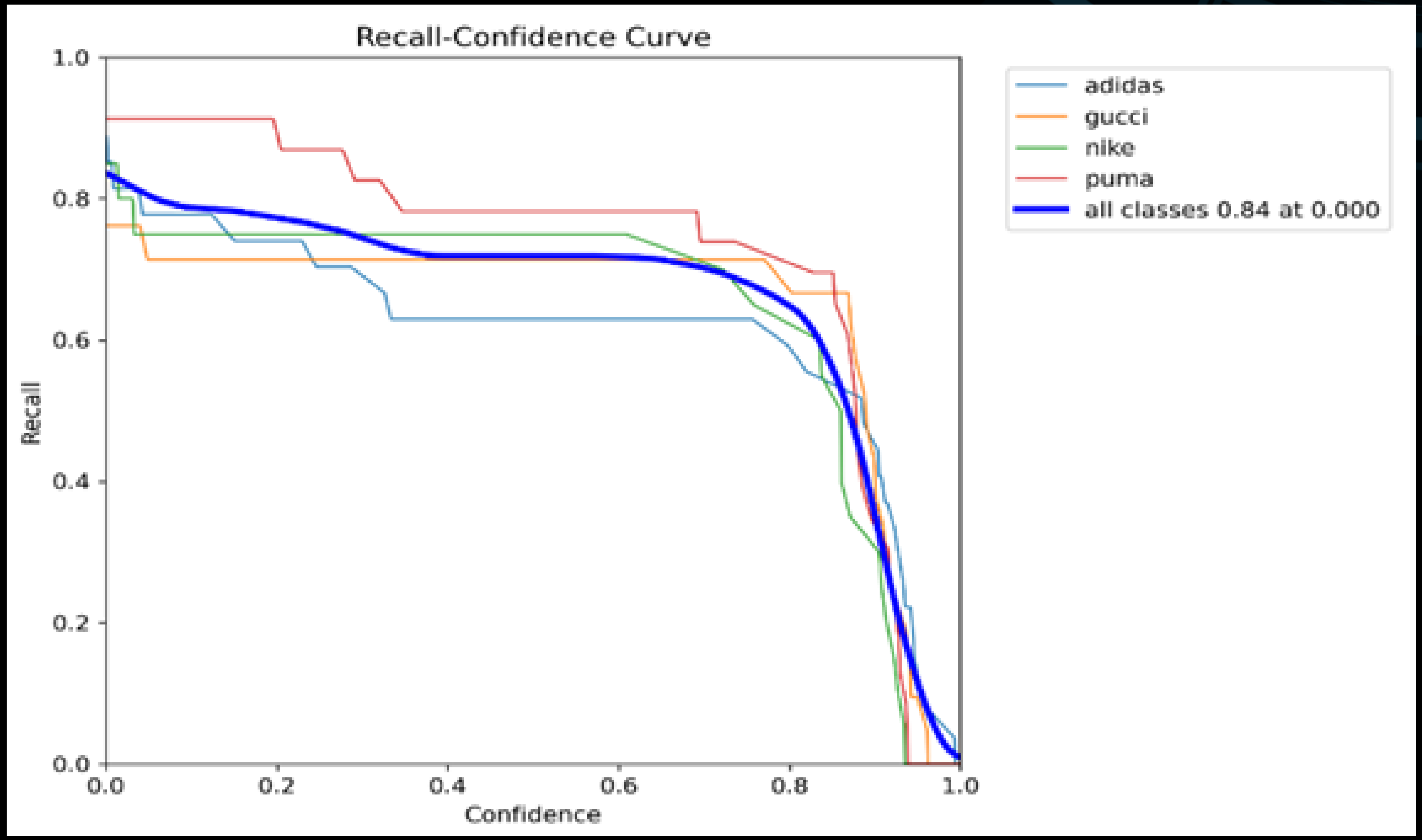
The F1-Confidence curve shows that the optimal operating region occurs around a confidence threshold of ~0.73, where the global F1-score reaches approximately 0.76, balancing precision and recall.

Figure 4 – Precision–Confidence Curve (Model-1)



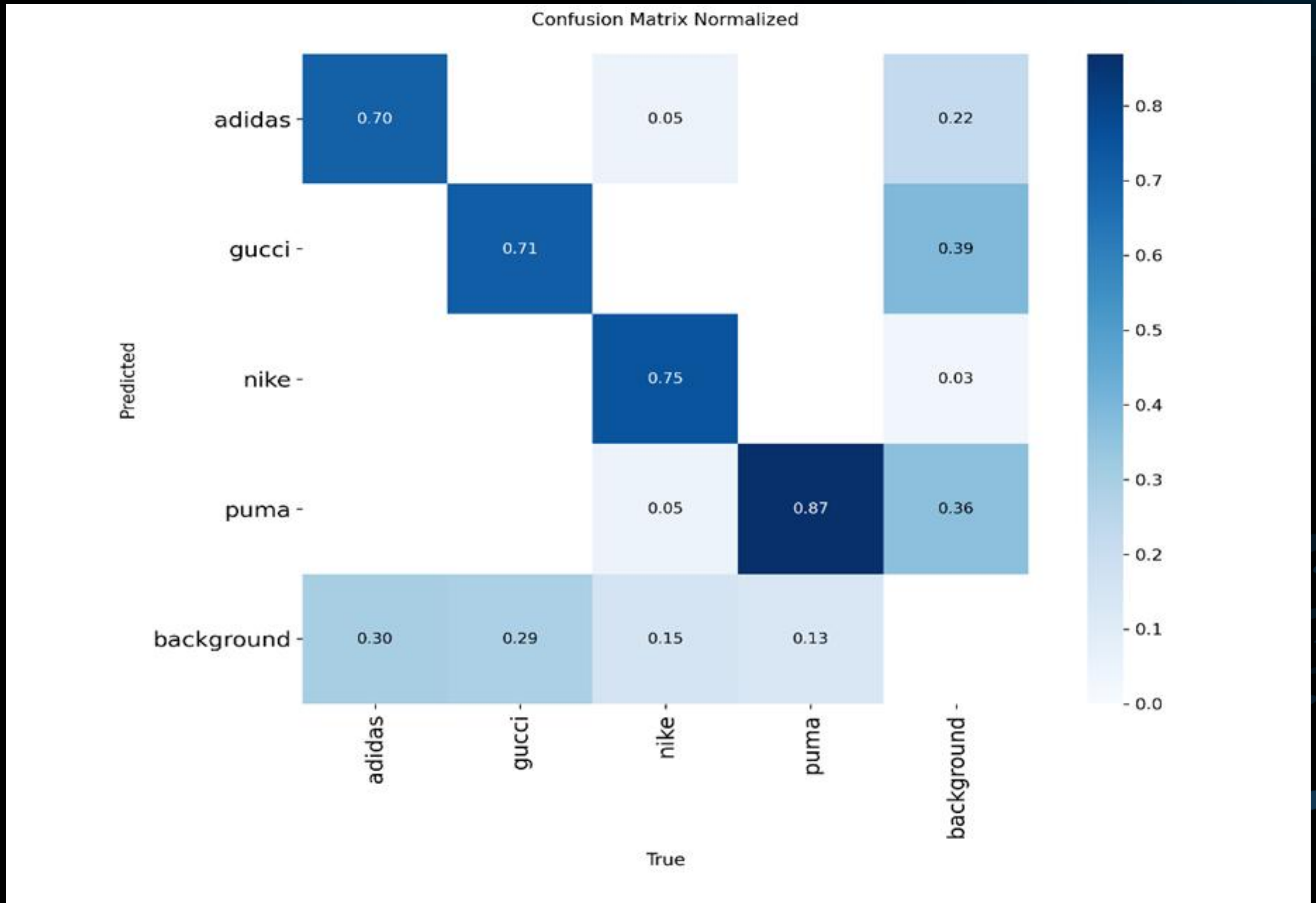
Precision increases as the confidence threshold rises and approaches very high values for highly confident predictions, indicating that Model-1 produces reliable detections when confidence is high.

Figure 5 – Recall–Confidence Curve (Model-1)



Recall gradually decreases as the confidence threshold increases, meaning that stricter thresholds reduce the number of detected objects. Lower thresholds therefore favour recall, while higher thresholds favour precision.

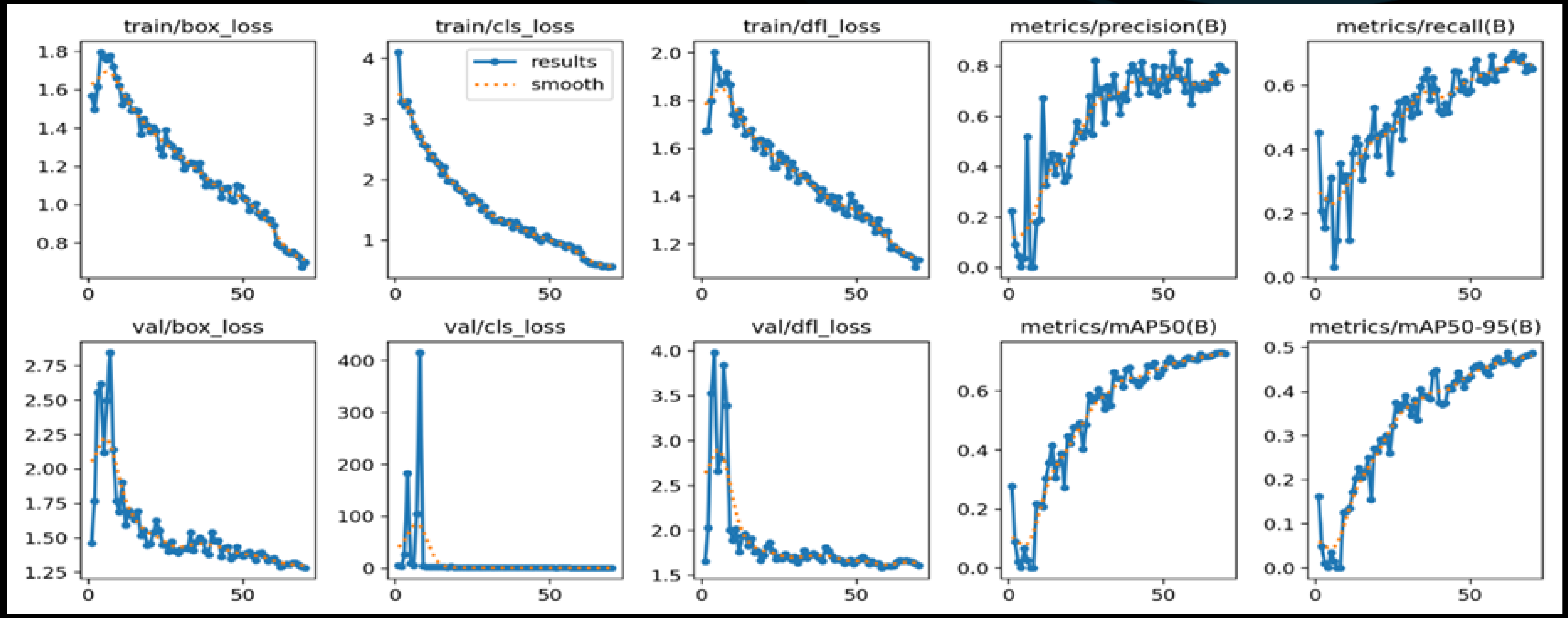
Figure 7 – Normalized Confusion Matrix – Model-1



The normalized confusion matrix confirms that Puma and Nike obtain the highest class-wise recall, whereas Gucci and Adidas experience more misclassification and background confusion.

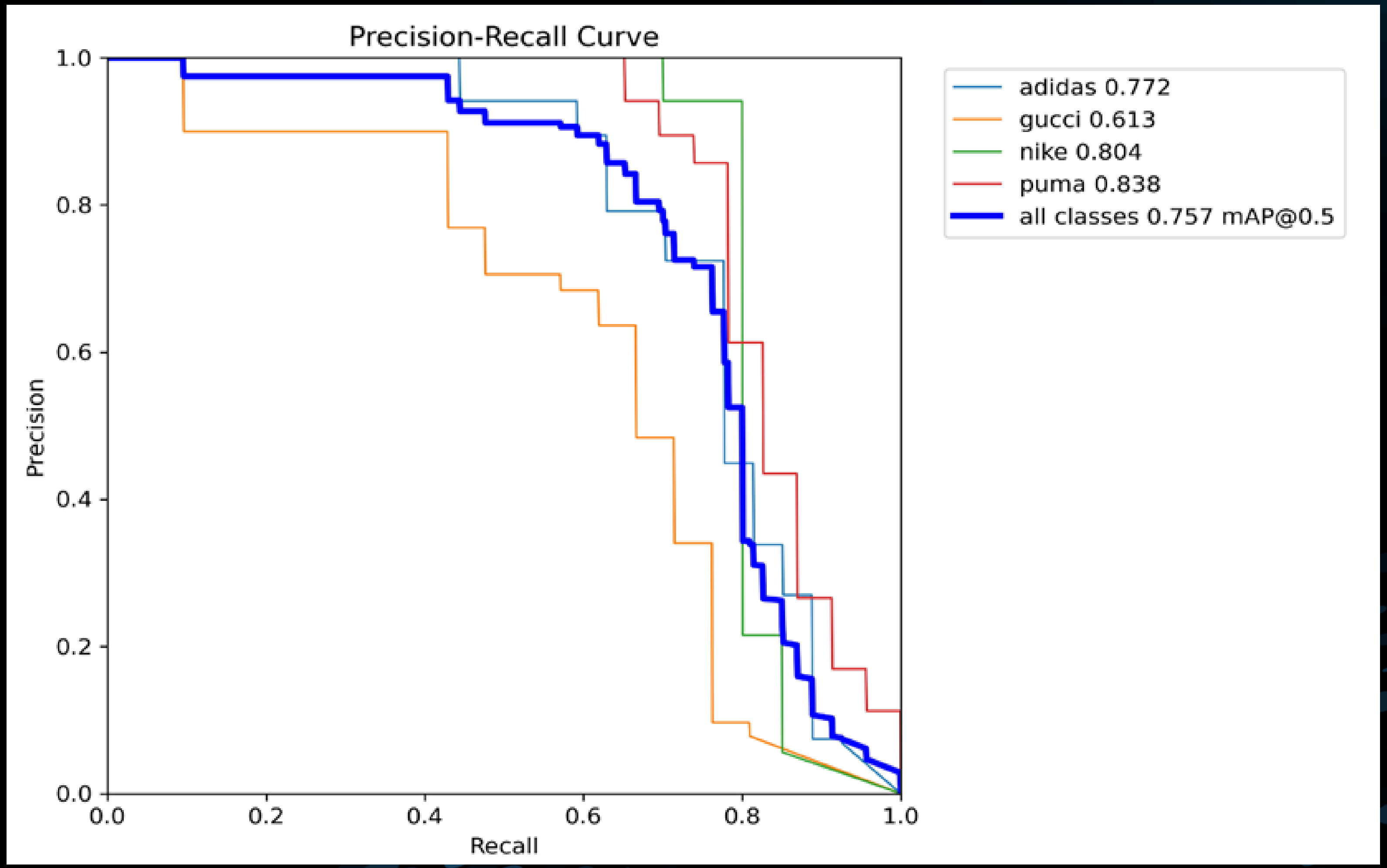
4.2 Model-2 : YOLOv8 Training Results:

Figure 1 : Training and Validation Curves for Model 2



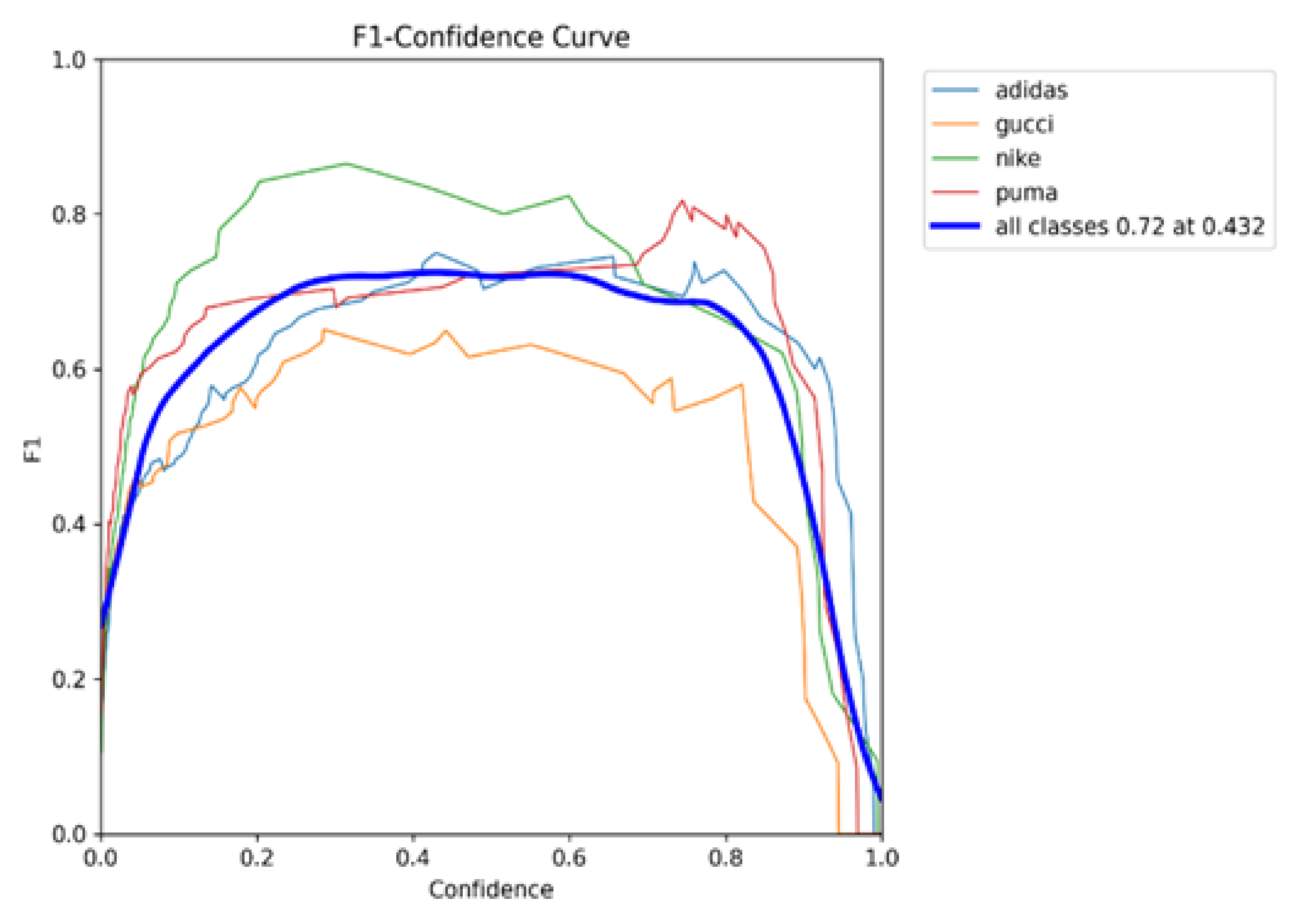
In figure 1 The training and validation losses (box, classification, and DFL) all decrease smoothly across epochs, showing stable convergence. Meanwhile, the performance metrics improve steadily: precision reaches about 0.85, recall rises to around 0.70, mAP50 reaches ~0.74, and mAP50-95 reaches ~0.50.

Overall, the curves show good learning progress without signs of severe overfitting, and the final mAP values indicate solid detection performance. Precision–Recall curves for Model-2:

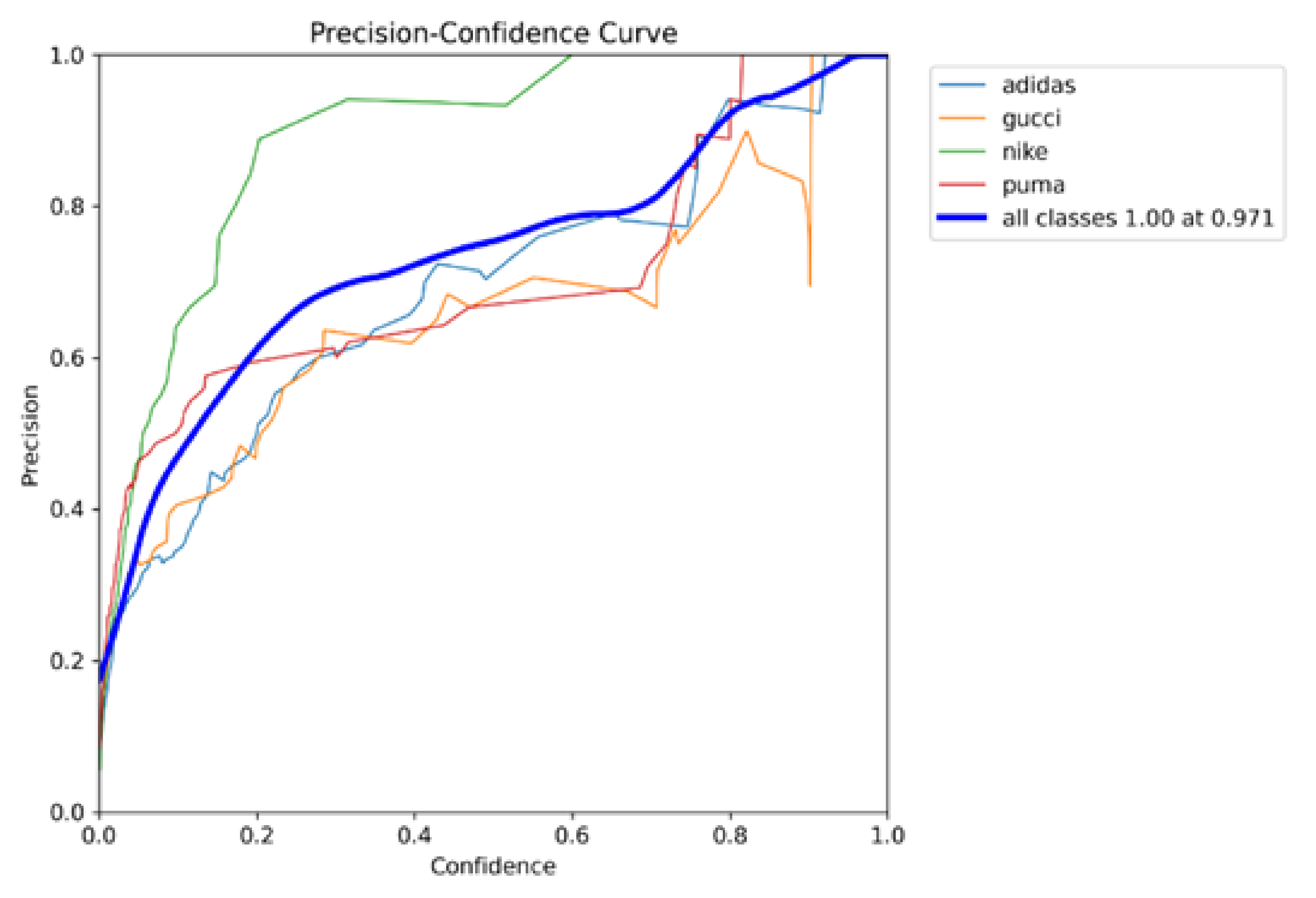


Model-2 achieves mAP@0.5 = 0.757, with Puma and Nike showing the strongest detection accuracy. The curves indicate that Model-2 maintains high precision across most recall values, demonstrating reliable logo detection performance.

3. Confidence–Based Performance :
F1–Confidence Curve:



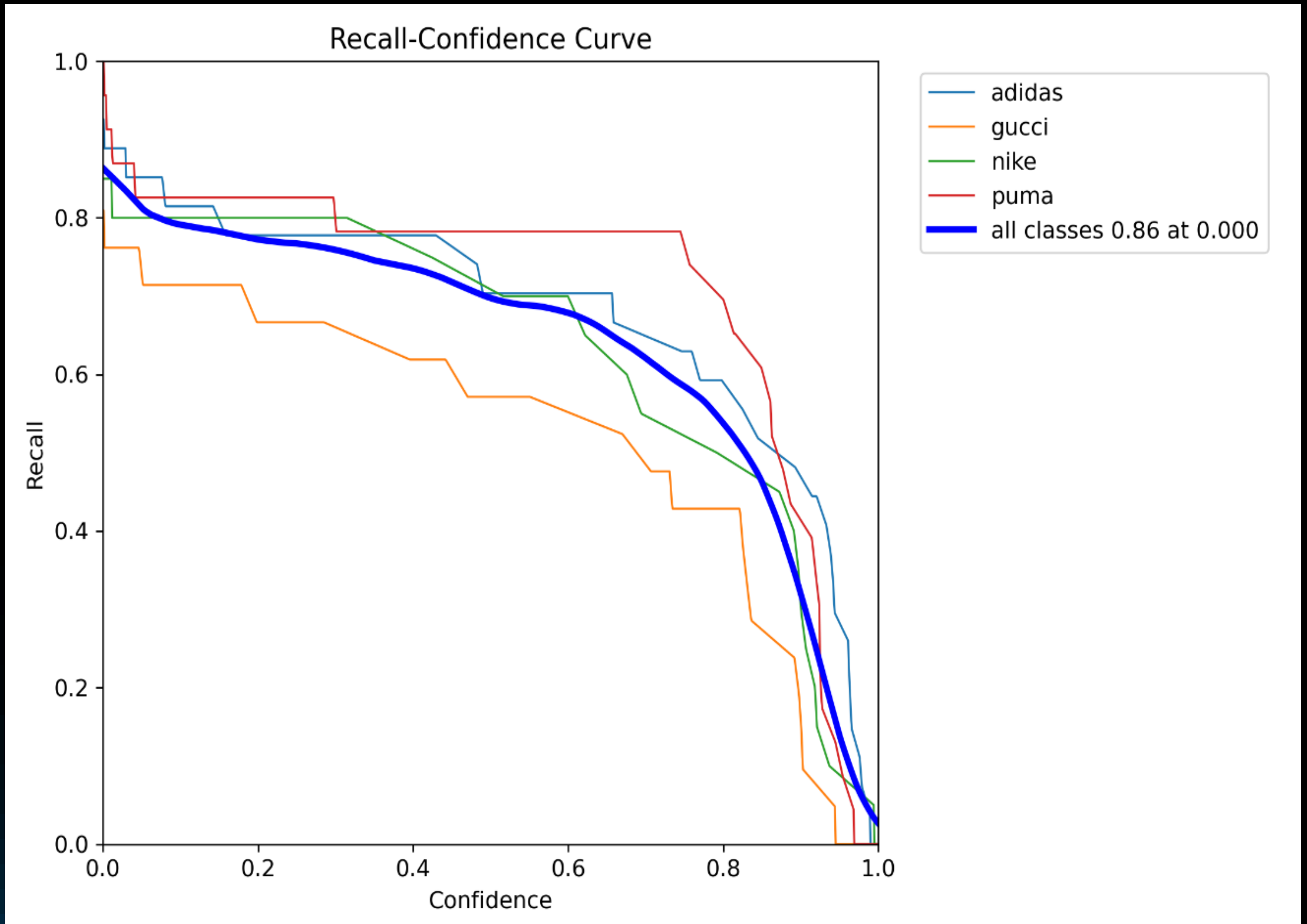
F1 vs Confidence
The F1-Confidence curve shows that the best performance occurs at a confidence threshold of about 0.43, where the global F1-score reaches ≈0.72. This threshold provides the best trade-off between false positives and false negatives.



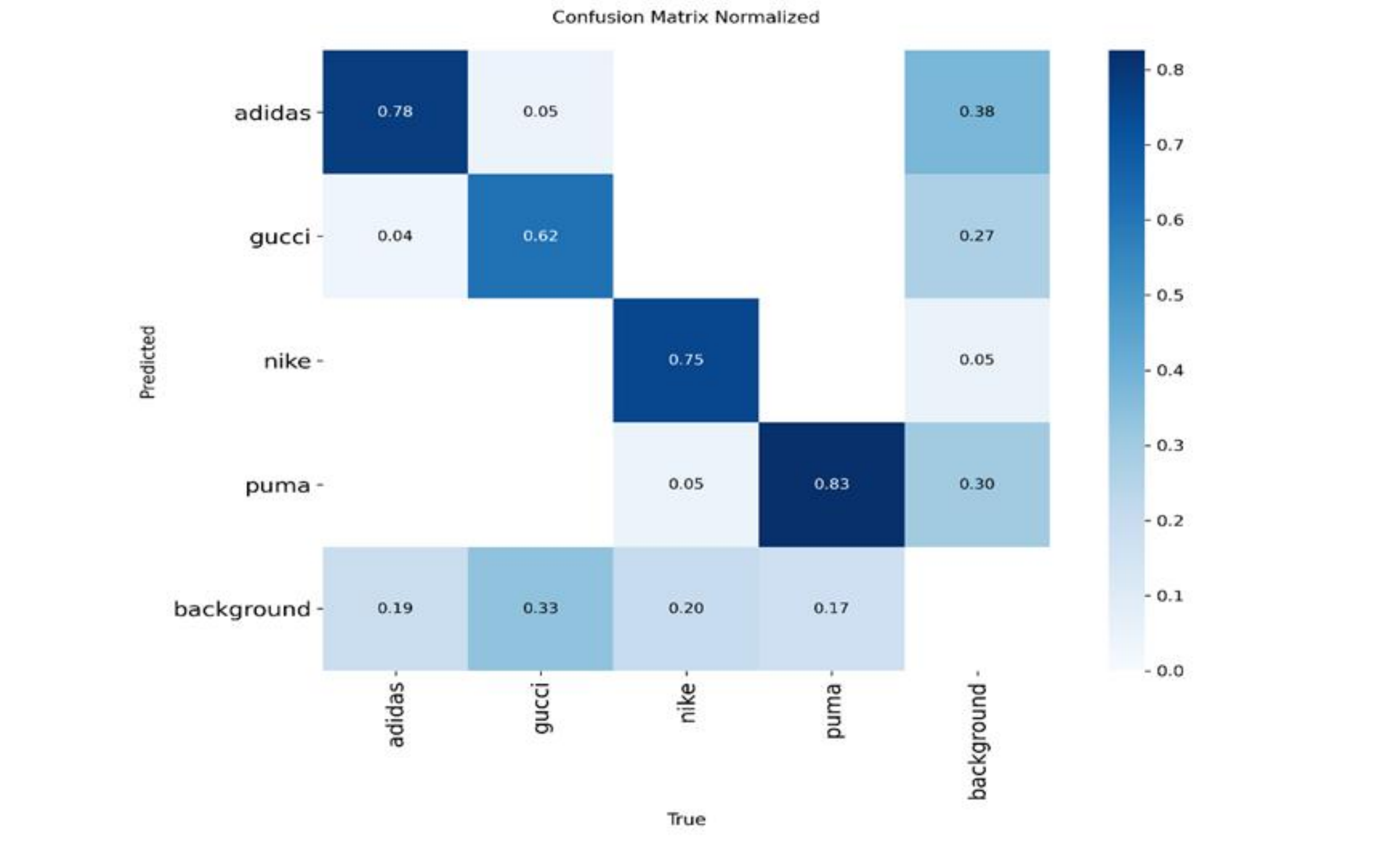
Precision vs Confidence
Precision increases as the confidence threshold rises and approaches very high values at strong confidence levels, meaning highly confident predictions are generally correct.

Recall vs Confidence

Recall decreases as the confidence threshold increases, since stricter thresholds remove more detections. Lower thresholds therefore maximize recall, while higher thresholds favor precision.



4. Normalized Confusion Matrix:



Model-2 correctly classifies most Puma (0.83), Adidas (0.78), and Nike (0.75) logos, while Gucci remains harder to detect (0.62). Some Adidas and Gucci instances are still confused with background, mainly when logos are small or unclear.

Model-2 detecting a Nike logo with confidence 0.93.

5. Comparison

Comparison Between Model-1 and Model-2

Model-2 (YOLOv8-Medium) clearly performs better than Model-1 (YOLOv8-Small) in this project. It achieves **higher precision, recall, and mAP (both mAP@0.5 and mAP@0.5–0.95)**, which means it detects logos more accurately, makes fewer false detections, and localizes logos more precisely. This improvement is also visible in the prediction images, where Model-2 gives **higher confidence scores** and more stable bounding boxes.

The main reason for this is that Model-2 uses a **larger and deeper neural network** with **higher input resolution (768 vs 640)** and **more training epochs**, allowing it to learn more detailed and complex logo features. This helps especially when logos are small, rotated, or printed on clothing with complex backgrounds.

Model-1, on the other hand, is **lighter and faster**, which makes it useful for real-time or low-power applications. However, it sometimes produces **lower confidence scores** and misses smaller or less clear logos.

A limitation for both models is that performance depends on the **dataset size and quality**. Some errors occur when logos are **very small, partially hidden, or blurred**, which shows that more data or better annotation could further improve accuracy.

Overall, **Model-2 is the better choice for accuracy**, while **Model-1 is better for speed and efficiency**.

And this is our github link where you can find our project work: https://github.com/youssefoueslati190-design/Logo-Detection-Project.git

THANK YOU

FOR YOUR ATTENTION