Logo Detection Using YOLOv9 on the Logo2K+ Dataset

MD Shah Imran Shovon

***Abstract*—This research report describes the use of YOLOv9 for logo detection with the Logo2K+ dataset. The investigation addresses difficulties such as dataset structure, lack of annotation, and time limitations. Results show that with appropriate modifications, the YOLOv9 model can detect logos with a mean Average Precision (mAP) of 72% over 300 epochs. Future work will involve improving data annotation and utilizing more powerful models.**

***Index Terms*—Logo Detection, YOLOv9, Logo2K+ Dataset, Deep Learning, Computer Vision**

I. Introduction

LOGO detection is a crucial task in computer vision, with several applications including brand monitoring, copyright enforcement, and content moderation. Accurately detecting logos in photos can provide significant information and enable automation in various fields. This project employs the YOLOv9 model, an advanced object detection framework, to identify logos in the Logo2K+ dataset. This study evaluates the performance of YOLOv9 for logo detection and suggests areas for development, despite limitations such as the absence of annotations.

II. Dataset Preparation

A. Data Download

The Logo2K+ dataset was downloaded from the provided GitHub repository. Upon extraction, it was discovered that the dataset structure diverged greatly from the YOLO format and missing label files. To address this, all images were consolidated into a single folder for better organization. Label files were then created for each image, using the entire image as the bounding box with the format: class number, center x, center y, width, and height. For training, two sessions were conducted with 500 and 50 randomly selected classes out of the original 2341 classes. The total number of images for training and testing was 167,029 and 50,182, respectively.

B. Data Structure and Selection

Given the three-day timeframe, training a big model on the complete dataset was not viable. As a result, an initial subset of 500 classes was chosen for the first training session, totalling 38,065 photographs. In a later session, after adjusting some hyperparameters, a subset of 50 classes, which had 3,616 images, was chosen for a quick training session, yielding better results.

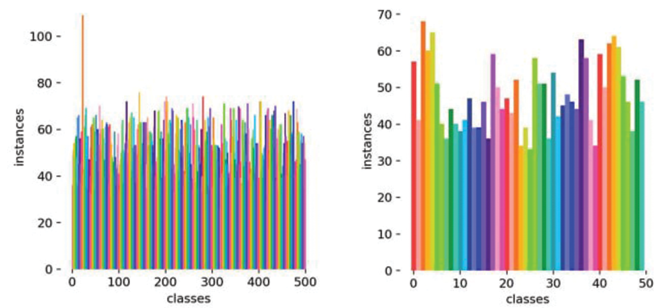

Fig. 1. Distribution of Images per class.

C. Annotation Script

To prepare the data for YOLO training, a script was developed to generate bounding box annotations. Due to the lack of specific annotations, the entire image was used as the bounding box (class number, center x, center y, width, and height) for each logo, simplifying the annotation process but potentially reducing the model's performance. Images per class were in between 30 to 70 range.

III. Model Selection and Training

A. Model Choice

The YOLOv9-tiny model was selected for its balance between speed and accuracy, making it suitable for the limited training time available. The table below presents a comparison among various Yolo models.

Model	Param.	FLOPs
YOLOv9-M	2.0M	7.7G
YOLOv9-M	20.0M	76.3G
YOLOv9-E	57.3M	189.0G

TABLE I
Comparison among various YOLO models.

B. Hyperparameters

The initial training was conducted with a batch size of 8 and default YOLOv9 hyperparameters. After noticing a plateau in the mean Average Precision (mAP) beyond 60 epochs, followed by fluctuations, the following adjustments were implemented:

- Zeroing out mosaic, mixup, and copy-paste augmentations.
- Increasing the model's depth and layer channel multiples.
- Increasing the Batch size to 32

These changes led to improved training performance, with mAP reaching 72% by epoch 300.

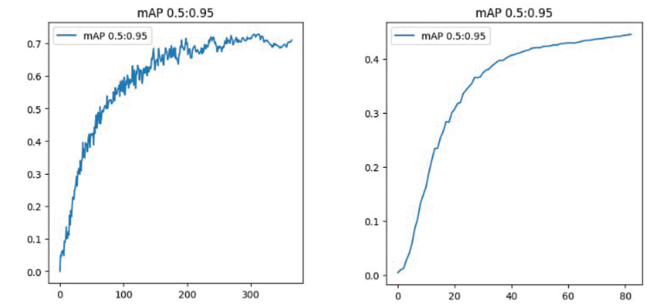

Fig. 2. mAP vs Epoch.

IV. Results and Observations

During initial training with the full 500 classes, the mAP reached 42% but then plateaued. I suspected that poor quality annotations and augmentations, which often selected plain parts of the image as objects, were causing significant performance issues. Therefore, I adjusted the augmentations and reduced the number of classes to improve the training efficiency. This tweak resulted in a significant performance gain, with the mAP reaching 72%. Furthermore, the model's learning ability was likely restricted by the minimal number of images or occurrences per class, which ranged between 30 and 70 for the majority of classes.

Fig. 3. Training Results for 500 classes

V. Discussion and Future Work

The structure and lack of annotations in the Logo2K+ dataset posed significant challenges for training with YOLO. Future work should focus on:

- Gathering more detailed annotations for better training data.

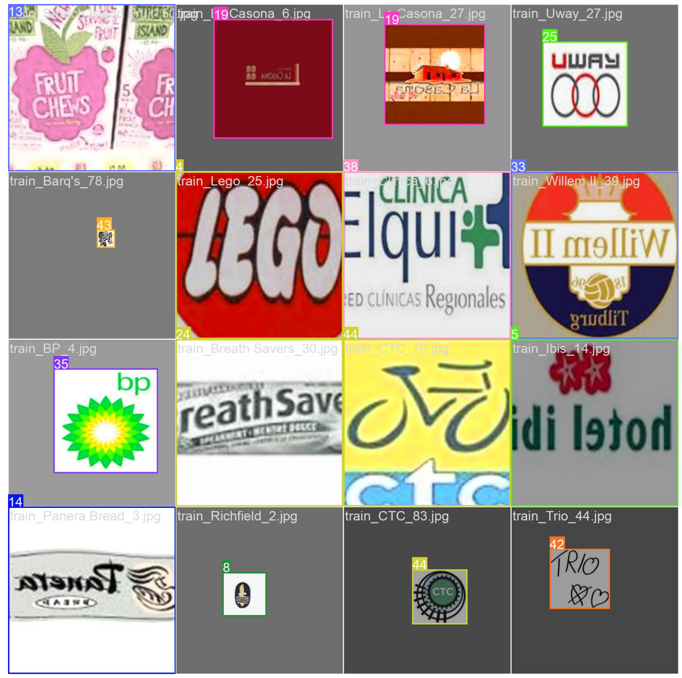

Fig. 4. Training Results for 50 classes

- Using a full-size YOLO model for enhanced performance.
- Implementing comprehensive data augmentation techniques to improve model generalization.

VI. Conclusion

This investigation demonstrates the use of YOLOv9 for logo detection on the Logo2K+ dataset. Despite the limitations, considerable insights were gained, emphasizing the need for improved data preparation and more rigorous training procedures. Further research should be conducted to improve the accuracy and reliability of logo detection tasks.

References

[1] WongKinYiu, "YOLOv9," Available: https://github.com/WongKinYiu/yolov9

[2] MSN199959, "Logo2K+ Dataset," Available: https://github.com/msn199959/Logo-2k-plus-Dataset