Classification on Logo-2K+ Dataset using YOLOv9

Fahim Hafiz

***Abstract*—"Logo-2K+" is a large-scale logo dataset with a diverse range of logo classes from real-world logo images containing 167,140 images with 10 root categories and 2,341 categories. This dataset is larger than the other existing datasets on logos such as FlickrLogos-32 and LOGO-Net. In this work, we apply the newly released sota (state-of-the-art) object detection algorithm YOLOv9 for logo classification purposes in the "Logo-2K+" dataset. The YOLOv9 classification model achieves 0.7 accuracy on the test datasets considering top5 accuracy.**

All the codes and necessary links can be found at the following GitHub Repository: https://github.com/FahimHafiz/Classification-on-Logo-2K-Dataset-using-YOLOv9

***Index Terms*—Logo Datasets, Image processing, Computer Vision, YOLOv9.**

I. Introduction

Logo-2K+ datasets [1] consists of 10 root categories: Food, Clothes, Institution, Accessories, Transportation, Electronic, Necessities, Cosmetic, Leisure, and Medical under which 2,341 subclasses of logo exist. These datasets are well-curated, cropped to the specific logos portion, and resized to 256 by 256 pixels. The authors provided the datasets in the train and test part and also proposed a new deep-learning model to classify the logos [1]. They proposed [1] a Discriminative Region Navigation and Augmentation Network (DRNA-Net) to classify the logos, which performed better than other existing deep-learning models such as NTS-Net(ResNet-50), ResNet-152+Efficient+LS, etc,. On the other hand, YOLOv9 is a sota object detection and segmentation model which is more accurate than its previous models such as YOLOv7, YOLOv8 as well as models like RTMDet, PPMYOLOE, etc,. in terms of object detection and segmentation on diverse image datasets [2]. YOLOv9 enhances model learning capacity by integrating Programmable Gradient Information (PGI) and the Generalized Efficient Layer Aggregation Network (GELAN). In this work, we want to explore the classification capability of this model on the "Logo-2k+" dataset. As the present YOLOv9 model architecture has no classification module, we modify the object detection module for classification purposes and discuss the findings.

II. Methods

The YOLOv9 architecture presently allows in-built object detection and segmentation modules on custom datasets. However, it doesn't have classification modules yet. For custom training, YOLO has YAML (Yet Another Markup Language) files which are basically human-readable data serialization language. These YAML files contain different variants of the YOLO's model architecture. For our classification purposes, we take the object detection model "yolov9-c". This "yolov9-c" has a backbone segment that extracts features from the datasets and a head segment that makes final predictions. We modify the "head" section of the model and keep the "backbone" section of the architecture intact. The codes can be found in the GitHub repository.

We intend to classify the 2,341 sub-classes using our model. So, we restructured the datasets by removing the 10 root categories. The datasets are organized in the train and test folders where each folders consist of all the 2,341 subclasses.

III. Results

We first trained the model considering 48 classes from the 2,341 logo classes. This is to see how the YOLOv9 works on a small portion of the whole dataset. Then, we again preprocessed the images from these 48 classes and performed augmentation to see if YOLOv9 works better when preprocessing is applied. After that, we intend to train the model on the overall datasets in similar way.

1) We see the confusion matrix and train, validation loss in 1 and 2.

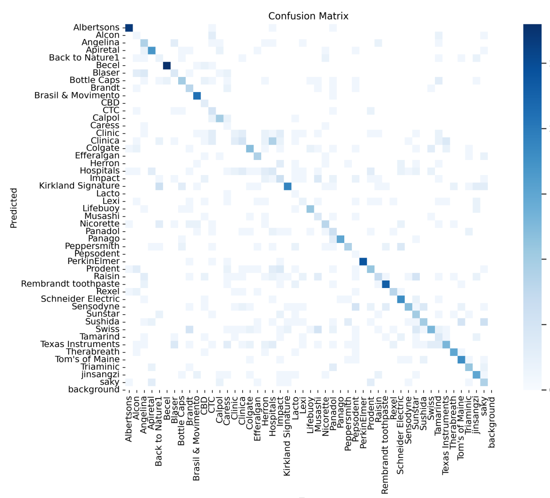

Fig. 1: Confusion Matrix for the 48 logo classes when trained with YOLOv9 (No Preprocessing on the datasets while training)

We see from 2 that the top1 and top5 accuracy are 0.49 and 0.7 when training the yolov9 without using any preprocessing such as data augmentation, filtering etc,. This approach achieves 3.54 test loss after 50 epochs. We also see an example of the predictions for such training in 3. We see that the model can predict correctly around 50% which indicates that the model is not good.

2) We preprocessed the data with augmentation, grayscaling, and high-pass filtering to see the effect on the outputs. We see the confusion matrix and train, validation loss in 4 and 2

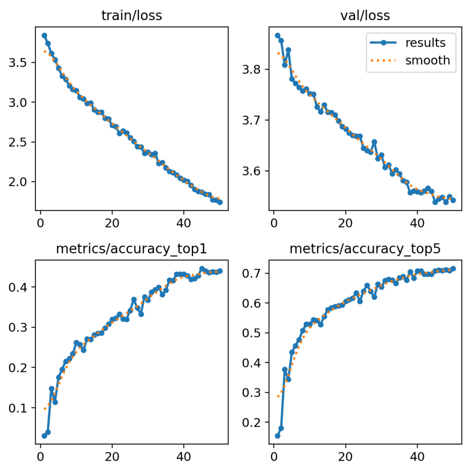

Fig. 2: Train loss, validation loss and accuracy for the 48 logo classes when trained with YOLOv9 (No Preprocessing on the datasets while training)

Fig. 3: Example of the Predictions when trained with YOLOv9 (No Preprocessing on the datasets while training)

for this case. We see from 5 that the top1 and top5 accuracy are 0.49 and 0.6 when training the yolov9 without using any preprocessing such as data augmentation, filtering etc,. Also, the test loss is 3.51 after 50 epochs which is lower for this approach compared to before.

Additionally, This model can predict correctly around 60% which is better on the test sets compared to before.

IV. Conclusion

The YOLOv9 took a long time to train on these 48 classes only. That's one of the reasons, we couldn't show the results for all the subclasses. Additionally, data preprocessing achieved lower accuracy. With time, we could achieve higher performance with proper data preprocessing and augmentations. For future works, we will try to achieve higher classification performance and we also could apply YOLOv9 for multilabel classification for a single image, that is we may classify each image with the root category and its main classes.

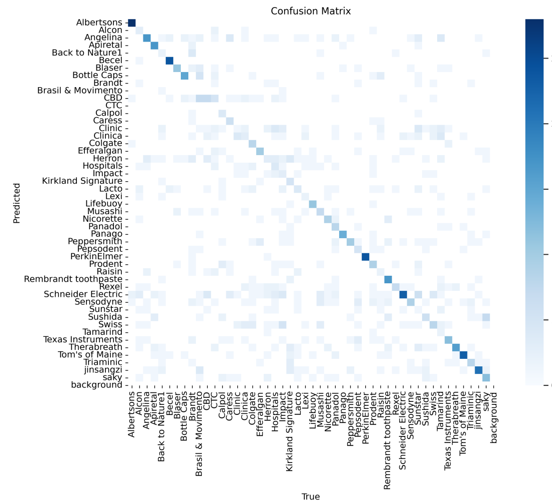

Fig. 4: Confusion Matrix for the 48 logo classes when trained with YOLOv9 (Data Preprocessed on the datasets while training)

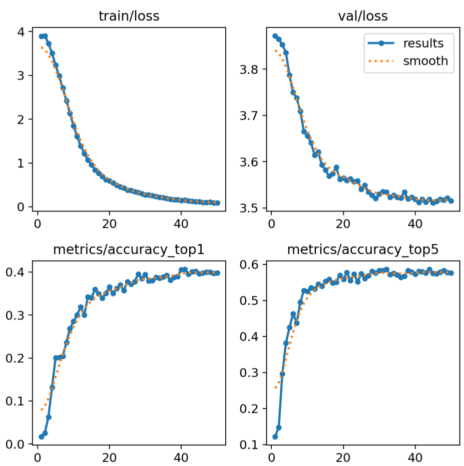

Fig. 5: Train loss, validation loss and accuracy for the 48 logo classes when trained with YOLOv9 (Data Preprocessed on the datasets while training)

References

[1] Wang, J., Min, W., Hou, S., Ma, S., Zheng, Y., Wang, H., & Jiang, S. (2020, April). Logo-2k+: A large-scale logo dataset for scalable logo classification. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 04, pp. 6194-6201).

[2] Wang, Chien-Yao, I-Hau Yeh, and Hong-Yuan Mark Liao. "Yolov9: Learning what you want to learn using programmable gradient information." arXiv preprint arXiv:2402.13616 (2024).