Adding Easy-to-use Fine-Tuning of Deep Learning Models with EVA

Hui Li
hli826@gatech.edu

Yiheng Mao
ymao310@gatech.edu

1 INTRODUCTION

EVA is an open-source AI-relational database with support for deep learning models. It aims to support AI-powered database applications that operate on both structured and unstructured data with deep learning models. The database has built-in support for object detection system YOLO which is widely used. However, at the current stage, users often prefer to use their own fine-tuned configurations instead of the vanilla model the database provides (Gaurav et al.).

2 RELATED WORK

YOLO is a unified object detection model. It is simple in construction and can be trained directly on full images. Unlike classifier-based methods, YOLO utilizes a stochastic grandient descent learning model trained on training loss, and the entire model is jointly trained. We use YOLOv5, a popular iteration of YOLO, as our primary object detection model (Redmon et al.).

Roboflow is a computer vision pipeline that enables developers to create pipelines for their CV applications. It provides necessary tools to train object detection models on custom data. In the context of our paper, we will use Roboflow's auto-annotation functionality to transform raw images and videos to labeled datasets trainable by YOLOv5 (http://www.roboflow.com/).

3 PROBLEM STATEMENT

Currently, if users were to fine tune YOLO, they need to download the data, extract the labels from EVA or an external labeling library, fine tune YOLO locally and then submit their custom configurations to EVA. This process is time-consuming and such a complex procedure is pruned to errors. Our project aims to address this issue by extending the functionality of EVA to support ad-hoc fine-tuning for YOLO. With the added support, instead of going through a lengthy fine-tuning process locally, users can just enable fine-tuning for YOLO, and all will be taken care of by the database.

4 GOALS

Our 75% goal is to add support for simple fine-tuning of YOLO. The fine-tuned model may not have performance on par with a locally fine-tuned version but is easy to use since it's built in. Our 100% goal is to support fine-tuning with performance comparable to locally fine-tuned versions by the user. In addition to all above, our 125% goal aims to allow users to provide simple instructions or configurations for how they wish YOLO is to be fine-tuned on EVA. In this case, EVA will fine-tune YOLO with some user preferences.

5 METHODOLOGY

We will use Roboflow to implement auto-annotation for the custom dataset. The ground truth generated by Roboflow will be used to build a custom version of YOLOv5. We plan to integrate the custom version of YOLOv5 with EVA, so that it can use the custom model for specific datasets. Before integration, we need to figure out how EVA integrates YOLOv5 and how to extend its functionality. We are considering whether we can integrate everything into a pipeline to make these tasks more convenient. Currently, we are still using the GUI to work with Roboflow. The pipeline will include auto-annotation, custom model training, integration with EVA, and fine-tuning functionality. Ultimately, we plan to deploy this pipeline on a Google Cloud virtual machine to make it easier for users to use.

6 DETAILED PROGRESS

First, we looked into the source code for EVA to see how YOLO is being integrated. We then realized that EVA uses a pre-trained version of YOLO (trained on coco dataset) available on torch hub with predefined set of labels. This may not be ideal for some user. We have devised a basic workflow for how to train YOLOv5 on custom data. We used the traffic video featured on EVA as an example. We used Roboflow to convert the each frame of the video to images and auto-annotated the images. Labels generated by Roboflow will then be used as ground truth to train YOLOv5 on the images. We will then test our custom-trained YOLO model on test images. Below are examples generated by the custom-trained YOLOv5 model. The example image without labels corresponds to the top-left image in subsequent figures. Due to limited computing resources, we are only able to train the custom YOLOv5 model in 50 epoches with batch size of 16. Therefore, our detection results may not be better than that of a pre-trained YOLO model, which no doubt has been trained extensively.

7 VALIDATION AND EVALUATION

To validate, our team will test the built-in fine-tuned YOLO on all unstructured data intended for YOLO such as images and videos. To evaluate its performance, we will compare its accuracy and efficiency with a locally fine-tuned version. Ideally, we can ask the developers of EVA what fine-tuned configurations a typical user will submit and how a typical user fine-tunes the model locally. If that is not possible, we can collect the data via some user study. Worst case scenario, we will simulate the fine-tuning process.

8 RESOURCES

Software:
EVA database
YOLOv5 object detection system
Roboflow pipeline

Hardware: Computing resourses to train YOLOv5 (preferably cloud)
Datasets: Typical data on EVA that is used with YOLO.

Figure 1: Example image without labels

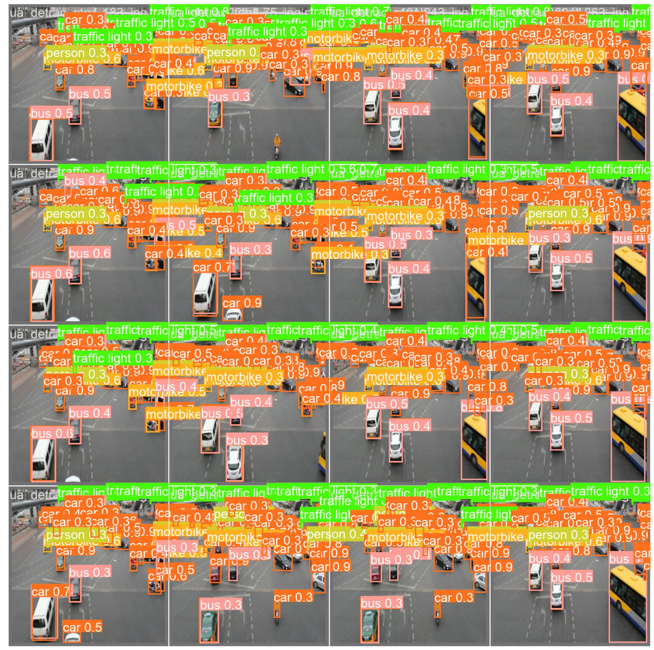

Figure 3: Example image with predicted labels

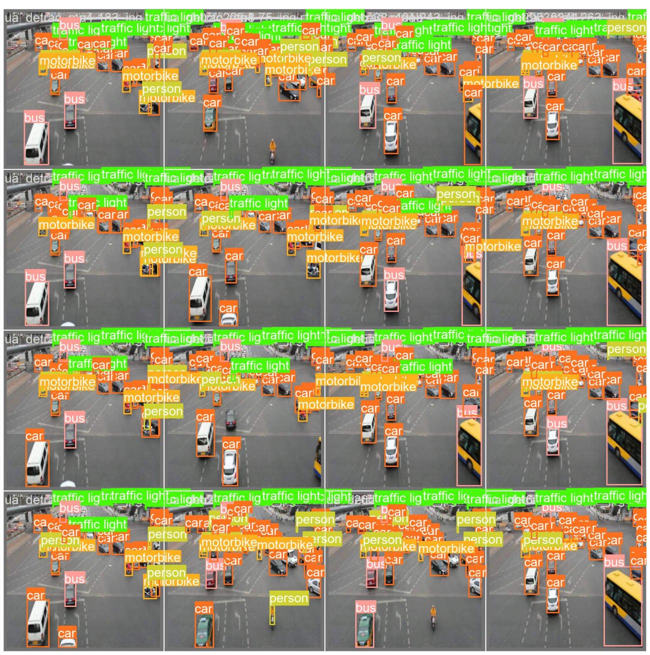

Figure 2: Example image with ground-truth labels

9 FUTURE WORK TIMELINE

Apr 2 - Apr 11: Enable visualization on detection results generated by user-trained YOLOv5 model. Moreover, visualization will be in video format if the original dataset is a video.
Apr 12 - Apr 20: Integrate the Roboflow pipeline with the local EVA environment and makes it more convenient for users. The pipeline will include auto-annotation, custom model training, integration with EVA, and fine-tuning functionality.
Apr 21 - Apr 24: Deploying the entire Roboflow and EVA pipeline to a cloud environment (such as Google Cloud VM) allows for greater scalability and accessibility.

10 PROJECT REPOSITORY ADDRESS

https://github.com/untrall/cs8803asi

BIBLIOGRAPHY

[1] Kakkar, Gaurav Tarlok, et al. "EVA: An End-to-End Exploratory Video Analytics System."

[2] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[3] “Give Your Software the Power to See Objects in Images and Video.” Roboflow, http://www.roboflow.com/.