CARZAM

CS 6235 – Real-time Systems

Project Final Report Fall 2022

Abrar Ahmed Mohammed

I. Deliverables

- Vehicle Data loading and Pre-processing
- Design & Implementation of ML Classifier using ResNet
- Metric collection and reproducing the work.
- Minimize error in classification when compared to ground-truth images
- Presentation
- Final Report

YouTube link containing presentation video:

https://youtu.be/J1O3iamRnVY

GitHub link for source code and supplement material:

https://github.com/AbrarAhmed647/CarZam--A-Vehicular-Classification-using-EDNA-ML.git

II. Introduction

The surveillance system is used for a variety of purposes, including vehicle tracking and real-time traffic monitoring. The system can be used by police officers for searching purposes, such as outlaw's vehicle identification in a crime. Officers typically manually identify the vehicle in recorded video based on its appearance. Although this method is accurate, it is time-consuming and prone to errors due to human fatigue for long-duration videos. Hiring employees is also expensive. Several machine learning methods, such as Fuzzy Logic, Decision Tree, Adaboost, Random Forest, Neural Network, and others, have recently been developed for vehicle classification. Convolutional Neural Network (CNN) is another example of such a method. CNN is a type of Deep Learning that falls under the neural network category. Because of its effectiveness, the method is now widely used in image recognition. There are two vehicle characteristics in the proposed vehicle classification: types and colors. Types are divided into four classes, whereas colors are divided into seven classes. CNN is then used to categorize vehicle images. CNN can achieve high performance in real-world applications, according to the experimental results.

The problem statement is, given the labelled data, classify the datapoints based on the similarity of the features. This project does the image classification for vehicles based on features like brand, type, and color. The vehicle image dataset is a large-scale data, it requires organizing and baselines of the dimensions of the dataset. EdnaML is a framework used for this task and it implements Residual-Net (ResNet) for the task of both single-branch and multi-branch classification. Machine/Deep learning workflows are difficult to manage, configure, and reproduce manually. EdnaML is a framework we use for running experiments and tracking results that gives us more control on the workflow.

EdnaML defines 2 pipeline abstractions:

Experiment Execution: Here, an ML model is trained on some training data, and evaluated on some corresponding test data. We perform experiment executions with ednaml.core.EdnaML

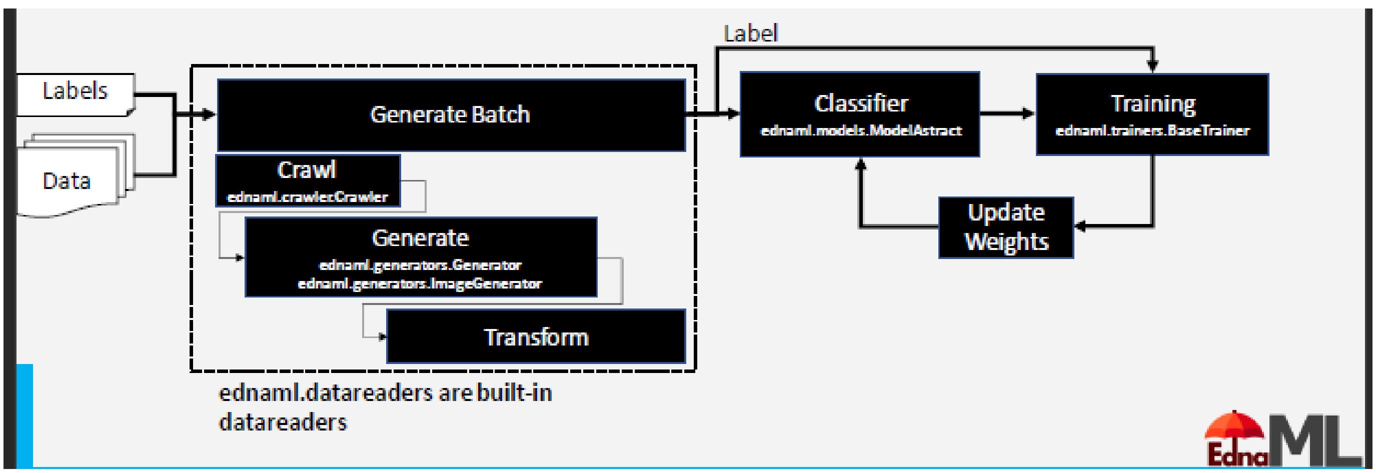
Model Deployment: Here, a trained ML model is used for some predefined task, such as unseen data labeling, supervision, or as a hosted service. We perform deployments with ednaml.core.EdnaDeploy

III. Related Work

Several studies have used CNN as a classifier in vehicle color classification. Chen et al. [1] proposed using feature context to identify the color of a vehicle. They used the classification with a dataset of 15,601 vehicle images divided into 8 color classes. They were able to achieve 90.68% classification accuracy. The authors of [2, 3] used CNN on Chen's dataset [4]. Their accuracy rates were 94.47% and 94.6%, respectively. Su et al. [5] proposed a new CNN structure called Colornet that achieved 95.74% accuracy in their vehicle color classification experiment. Alexnet [6] and GoogleNet [7] were outperformed by the structure. Zhou et al. [8] proposed deep neural network or deep learning approaches for vehicle detection and classification in another paper. They employed YOLO [9] as a detection model. As classification approaches, Alexnet was used. There are four types of classification in classification modules: passenger vs other, cars vs vans, sedans vs taxis, and sedans vs vans vs taxis. Following the application of both structures, they were fine-tuned to be compatible with the public dataset provided in [10]. The main goal of the proposed method in this project is to improve the accuracy of the previously mentioned vehicle type and vehicle color classification. Moreover, formalize the work using EDNA ML. In this work, a convolutional neural network with two convolution layers is chosen as the classifier.

IV. PROPOSED WORK PIPELINE

The pipeline is categorized into two parts i.e., Data and ML. In the data section of the pipeline, we implemented a crawler that generates a list of data, a generator that generates batches of data, and a Transform to augment or transform images. Then a classifier is implemented to obtain predictions for the individual batch. Then, these predictions are compared to the ground truth and performance metrics such as accuracy and F1 score have been computed. The work has been formalized in the EDNA ML and helps us to easily track, change and reproduce the experiments. Therefore, it adds more control and structure to the Deep Learning workflow



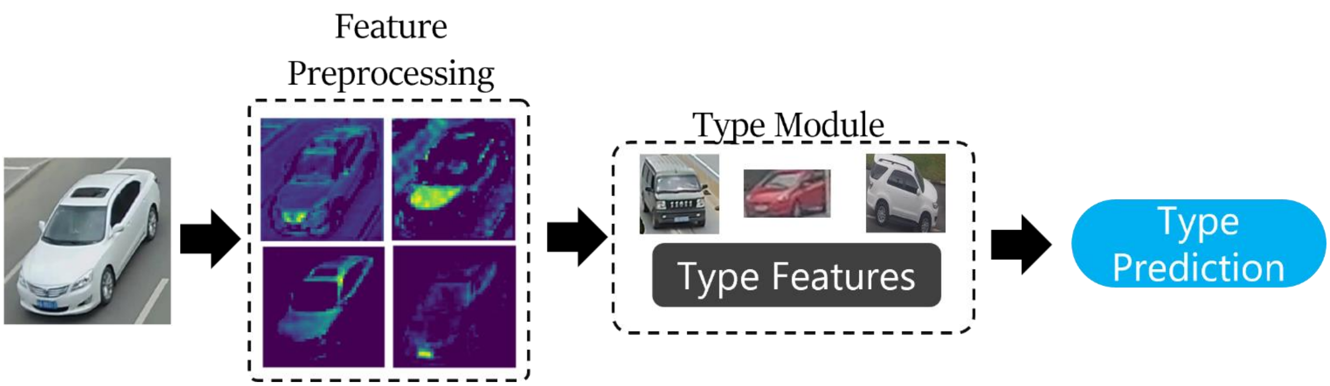
CRAWLER:

- Images in each directory are labeled with their type, year, make, model, and submodel
- Naming convention means make and model are combined together
- Crawler is implemented which generates a list of tuples which have the structure of [(path,type,color,year,make),…….n]
- Crawler has 2 components self.classes is a dictionary with information on classes and values and self.metadata which is a list of tuples data is where each entry is a data point
- Then, we built reverse lookup dictionary to convert words to numbers

CLASSIFIER:

- We used Classification Generator or MultiClassificationGenerator to process the data from crawler into a format that the ML model can use to transform predictions
- **Single Class Classification:**

The architecture looks like below for label 'type'



Similarly, Single class classification is applied to Vehicle color and make

- **Multi class Classifiers:**

Multiclass classifiers try to classify multiple things at once, using the same features. Sometimes it works, if the features are co-located or have some overlap.

V. RESULTS

	Type of Exp	Label	Accuracy	Micro F1 score	Weighted F1 score
Single class	Single vehicle type	TYPE	41.414%	0.414	0.321
	Single vehicle color	COLOR	76.768%	0.768	0.727
	Single vehicle make	MAKE	53.030%	0.530	0.477
Multi class	Multi-class Color-Type	Color	74.2%	0.742	0.701
		Type	31.3%	0.313	0.259
	Multiclass Color-Type-Make	Color	72.2%	0.722	0.675
		Type	36.9%	0.369	0.328
		Make	21.7%	0.317	0.223
Multi Branch	Color+type->Make	Color-fc	70.7%	0.707	0.650

	Type of Exp	Label	Accuracy	Micro F1 score	Weighted F1 score

Multi Branch	Color+type->Make	Type-fc	37.9%	0.379	0.295
		Fuse	23.2%	0.232	0.192
		Color-branch	10.6%	0.106	0.065
		Type-branch	15.2%	0.152	0.120