Internship Experience - [İkon Arge - Car Brand Classification]

Şerif Can Tekin

8 November 2023

1 About the Project

During my internship, I worked on a project aimed at classifying car brands using machine learning. The goal was to develop a model that could accurately determine the brand of a car based on its image. This task has practical applications in areas like automotive marketing, inventory management, and automated vehicle recognition systems.

The project involved collecting a dataset of car images, preprocessing the data, and training machine learning models to classify car brands. By experimenting with different algorithms and techniques, I aimed to create a reliable model capable of accurately identifying various car brands from input images.

2 Stages of the Project

2.1 Preparing and Customizing an Object Detection Model

I prepared an object detection model tailored to our project's needs. This involved researching and implementing object detection algorithms suitable for identifying cars within images. I experimented with adapting TensorFlow's frameworks for object detection to our specific use case.

2.1.1 Download TensorFlow Repo

The first thing we need to do is download the tensorflow repo from the github link. https://github.com/tensorflow/models

2.1.2 Protobuf/Protoc Download

We need to use Protobuf. Protocol Buffers (Protobuf) is a data format and language developed by Google for cross-platform data serialization and communication. You can download from here: https://github.com/protocolbuffers/protobuf/releases. You can add protoc to the path.

2.1.3 Anaconda

In this part, we need to use anaconda. We will create a new environment. Then, we will install protobuf.

2.1.4 Proto

I left a file named protoBuf.py in the repo. (You can also access this file from the following link: https://github.com/TannerGilbert/Tensorflow-Object-Detection-with-Tensorflow-2.0). This Python code compiles the .proto files in the specified directory using the Protocol Buffers compiler and generates Python-specific data classes. These classes are then saved in the current working directory. As a result, you can utilize Protobuf messages in your Python program by including and using these generated data classes. I will put this python file to the TF2/models/research. The file needs to be compiled. It will be like that:

python protoBUF.py pathtoDirectory pathtoProtocFile

2.1.5 Installation

There is a python file that has name setup.py in directory research/object_detection/packages/tf2. We will copy this file under research folder. This file will help us install the python package and necessary dependencies.

python -m pip install .

conda install numpy

We complete the operations regarding the environment with these two codes.

2.1.6 Download the Model

wget is a Python module used to download files from URLs. This module makes it easy to download files from the web and provides simple usage. That's why we need to install this library. Firstly, we should move modelDownloader.py under object_detection folder. Then I will open terminal in my IDE, activate our environment in conda like that:

conda activate "environment_name"

Then we install wget library:

pip install wget

Now everything is ready, we can download our model using modelDownloader.py file. You can also choose a different model from here:

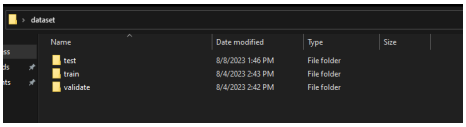
github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

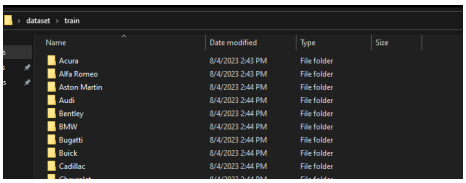
2.2 Data Preprocessing

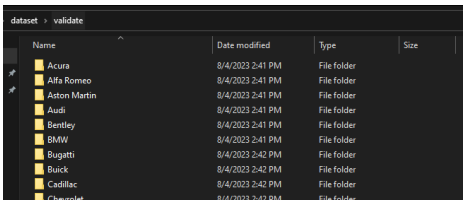
During the project, I focused on preparing the car image dataset for training. By ensuring the dataset's quality and diversity, I laid a solid foundation for the subsequent training of machine learning models.

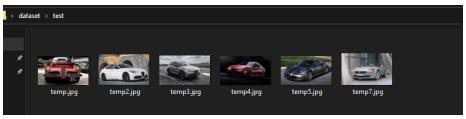
2.2.1 Data Format

First of all, I need to show what form we want the data to be.

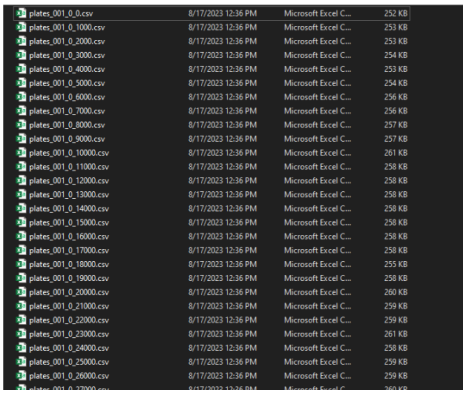








Now, I want to describe the format of the data we have. The data we had came in two forms. The first was jpg files and the second was csv files. Unfortunately, I do not have images of the jpeg files at the moment. On the other hand, csv files was like that:



	A	B	C	D	E
1	0	KAMAZ	0	0	http
2	1	Mercedes-Benz	Viano	0	http
3	2	Schmitz	0	0	http
4	3	Schmitz	0	0	http
5	4	Volkswagen	Sharan	0	http
6	5	Chrysler	300C	0	http
7	6	Chrysler	300C	0	http
8	7	Chrysler	300	1st gen, 2005-2010	http
9	8	Chevrolet	Lumina	2nd gen, 1994&2001	http
10	9	Volkswagen	Passat	0	http
11	10	Alfa Romeo	156	(932), 1997-2003	http
12	11	Volkswagen	Transporter	0	http
13	12	Mercedes-Benz	E-Klasse	0	http
14	13	Mazda	626	0	http
15	14	Ford	Mondeo	0	http
16	15	KamAZ	55111	0	http
17	16	BMW	X5	0	http
18	17	SEAT	Alhambra	0	http
19	18	License plate without vehicle	0	0	http
20	19	Volkswagen	Transporter	0	http

2.2.2 Running

Then, I put the file named DetectionAndOptimization.py, which I gave in the repo, under the object detection (model/research/object_detection) folder. I put my csv files in this directory: model/research/object_detection/inputs/csv_files Now I come to the terminal of my IDE, go to the object_detection directory and then run the following code:

python .\DetectionAndOptimization.py -m
ssd_mobilenet_v2_320x320_coco17_tpu-8\saved_model -l .\data
\mscoco_label_map.pbtxt

Now we have the data we will need to develop the model.

2.3 Model Training

In the training phase, I tried for the development of machine learning algorithms for car brand classification. Leveraging various algorithms and fine-tuning techniques, I aimed to create a robust model capable of accurately identifying car brands from input images. Through meticulous experimentation and analysis, I worked for the creation of a highly accurate classification model that met the project's objectives.

2.3.1 Preliminary Preparations

First we need a Python environment. For this, we will use conda again and create an environment. Then we will activate the environment and install the following packages into it.

- python
- numpy
- tensorflow
- opencv-python
- pillow
- matplotlib
- SciPy

We had already created our dataset before. Our preparations have been completed.

2.3.2 Training

For this part, I will use a model builder that I found on github. Here the github link:

github.com/feitgemel/TensorFlowProjects/tree/master/transfer-learning/Resnet50-CarDetection

Here important part, we should arrange our interpreter. I need to use the environment that I create in conda. I just need to do some changes in the files TransferLearn-01-BuildTheModel.py and TransferLearn-02-TestTheModel.py. I will arrange the paths according to the my computer. Now, everything is ready. I will come to terminal of IDE, activate the environment. Then, I will build the model:

python TransferLearn-01-BuildTheModel.py

Now, our model is trained. I will come to TransferLearn-02-TestTheModel.py file, and give the path of our trained model to the file. Also I will give the path to which image we want to be tested. Then run. That's all.

python TransferLearn-02-TestTheModel.py

3 Technologies and Tools

Some of the key technologies and tools I utilized while working on this project include:

- Python programming language for data preprocessing, model training, and evaluation.
- Scikit-learn library for implementing and evaluating machine learning algorithms.
- Conda package manager for managing Python environments and dependencies.
- NumPy library for numerical computing and array operations.
- OpenCV library for image preprocessing and augmentation.
- Matplotlib library for data visualization and analysis.
- Pillow library for image processing tasks such as opening, manipulating, and saving images.
- SciPy library for scientific computing and statistical analysis.
- Jupyter Notebooks for interactive development and experimentation with machine learning models.
- TensorFlow libraries for implementing and training neural networks.

4 How to Contribute?

If you're interested in contributing to this project, please. Your feedback and contributions are valuable to me. serif.tekin@metu.edu.tr