

COMPS492F Machine Learning

Car Classification Project (10% of the final grade, 33.3% of OCAS)

Due Date: 15 April 2023 23:59

Aim

Machine learning is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. This project aims to let students become familiar with implementing machine learning projects. You will be asked to conduct a **car classification project** to perform the task of classifying cars with 20 different classes. Each group of **1-2 students** will submit an integrated python code that applies your model to solve the classification task.

Project Requirements

- Train a model that can perform car classification tasks using **Python**.
- Implement a **training script** (E.g., **train.py**) to perform the learning procedure and obtain a classification model.
 - You can **apply any techniques that perform classification tasks** in your training
 - Logistic Regression
 - K-nearest
 - Decision Tree
 - Random Forest
 - Naïve Bayes
 - CNN
 - etc.
 - You should **implement the validation step** after the training.
 - A **loss function** needs to be chosen for your measurement of the learning process
- **Tensorflow** or **Pytorch** framework are both allowed to apply to build this project.
- You can apply **transfer learning** during training.
 - E.g., YOLO
- the **basic procedure** of machine learning should be implemented, for example, data preprocessing, techniques application, training, validation, etc.
- In the last stage, an **integrated python script** (**test_model.py**) needs to be implemented which is used to apply your model to perform classification.
 - **You should implement python argument (E.g., argparse, sys.argv) with the script to enable users to set up some hyperparameters in the command line interface**
 - E.g., In the terminal, we can type like this to start training:
test_model.py [test data path] [trained model path] [saved result path]
 - **This script should have functionalities that can receive a testing folder input, apply your model, and save the classification results to a .txt file.**
 - A dummy script is provided.

Dataset

- A basic [dataset](#) will be provided to you, which has already been separated into the training set, validation set, and test set. (**Access Password: COMPS492F**)
- The dataset includes **20 classes of the type of car**, all in **.jpg format**, and a csv file containing the classes we want you to classify and the amount of data for each class.

number_of_samples					
	Name of class	Number of files	train	val	test
0	bmw serie 1	74	45	15	14
1	chevrolet spark	98	59	20	19
2	chevrolet aveo	117	71	23	23
3	clio	310	186	62	62
4	duster	252	152	50	50
5	Golf	145	87	29	29
6	hyundai i10	117	71	23	23
7	hyundai tucson	116	70	23	23
8	logan	101	61	20	20
9	megane	252	152	50	50
10	mercedes class a	76	46	15	15
11	nemo citroen	73	44	15	14
12	octavia	141	85	28	28
13	picanto	161	97	32	32
14	polo	119	72	24	23
15	sandero	234	141	47	46
16	seat ibiza	90	54	18	18
17	symbol	143	86	29	28
18	toyota corolla	136	82	27	27
19	volkswagen tiguan	166	100	33	33

Figure 1. csv file in dataset

- **You can enlarge your dataset with online resources according to your needs to improve the performance of the model.** But remember to save your own data on your computer because we need you to submit your dataset also.

Project Report

- A report of **2-3 pages** addressing the details of your project implementation
- In the written report, you should include:
 - Group/Individual basic information: name, student ID, tutorial session.
 - Description of the data preparation, classification implementation, and model testing details. Basically, what you have done during project implementation should be included in the report.

Submission

- Submit your works to OLE, which include:
 - A **runnable python script** which applies your model to perform car classification.
 - A **project report in pdf format**.
 - A **zip file** including all the necessary files in your project: dataset, training script, log files, etc.
 - A **.txt file** includes the commands you used for training (train.py), for the runnable python script mentioned above (test_model.py).

Grading Referenced Criteria

The grading is mainly based on the performance of the classification model.

- Pass Grade
 - **Satisfying the basics requirements** and obtaining at least **93% of accuracy in validation set.**
 -
- Higher Grades
 - The **integrity of your project implementation**, i.e., All necessary procedures should be implemented for your project, i.e., data preprocessing, techniques selection, training, validation, testing, measurement, etc. Show them on your report,
 - **Improve model performance** in terms of accuracy.