Grab Car Classifier

Creator: Lim Ji Chen

Date: 14/6/2019

Email: jasonakon@gmail.com

Introduction:

This project inspired by Grab "AiForSea", Computer Vision challenge to classify and recognise distinct car images based on a given dataset. Here, car make and models can be classified immediate with any input images that belong to the provided car class dataset family which are 196 in total. This project uses a lightweight model that allowed to produce faster result and accurate result at the same time.

Thank you for giving such opportunity and let's get ready to classify!

Dependencies:

You'll need have the following installed:

- Python3
- Opencv
- Tensorflow v1.13 (latest-recommended)
- Pillow
- Numpy
- CSV
- Windows/Ubuntu

Trained Model:

Model can be found at: "car_classifier /trained_model/grab_model/"

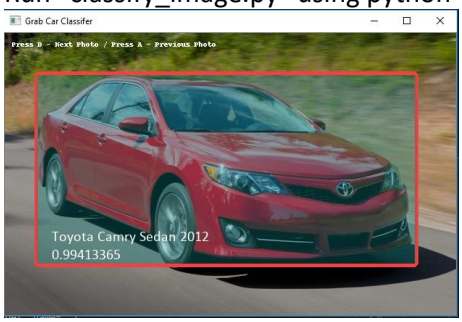
How to?

Head into working directory: "car_classifier/":

Then, this project provides **3 different methods** of classification based on your input preferences:

Method 1: Classify and view (images):

This method allows you to view the output generated from the model through a simple GUI:

1. Insert your test dataset (images) into the folder "test_images"
2. Run "classify_image.py" using python
3. 

4. To navigate your test photos using (keyboard):
    a. **Press D – Next Photos**
    b. **Press A – Previous Photos**
5. Observe the class prediction and its confidence level in GUI and command log.

**Note: sometimes you need to hold it or press multiple times to navigate*

Questions email at: jasonakon@gmail.com

Method 2: Classify into CSV (images): [Recommended]

This method allows you to generate all the prediction output based on your test image dataset into .csv format which include the image file name and 3 top class predictions and its confidence level % as well.

1. Insert your test dataset (images) into the folder “test_images”
2. Run “classify_image_into_csv.py” using python
3. Observe the command log and wait for it to generate all outputs
4. Once successful, proceed to view the “car_classify_output.csv”
5.

filename	predict_1	confidence_1	predict_2	confidence_2	predict_3	confidence_3
test_images\00032.jpg	Dodge Durango SUV 2012	0.74173677	Jeep Grand Cherokee SUV	0.3058213	BMW X3 SUV 2012	0.23162058
test_images\00033.jpg	Toyota Camry Sedan 2012	0.99413365	Toyota Corolla Sedan 201	0.025250494	Honda Odyssey Min	0.00967437
test_images\bugatti-veyron-2012-buga	Bugatti Veyron 16.4 Coupe 2	0.91737694	Bugatti Veyron 16.4 Conv	0.15415901	Acura ZDX Hatchbac	0.022095233
test_images\download.jpg	Lamborghini Aventador Cou	0.3325676	Aston Martin V8 Vantage	0.09009412	Lamborghini Gallard	0.088701576
test_images\IMG_15184-medium.jpg	Bugatti Veyron 16.4 Coupe 2	0.96608436	Lamborghini Gallardo LP 5	0.13847479	Bugatti Veyron 16.4	0.03625989
test_images\Red_Bugatti_Veyron_on_	Bugatti Veyron 16.4 Coupe 2	0.9808343	Spyker C8 Coupe 2009	0.11106351	Bugatti Veyron 16.4	0.06649193

Method 3: Car detection (videos):

This method enables its utility feature to predict output in real-time video playback:

1. Insert your test dataset (videos) into the folder “test_videos”
2. Run “classify_video.py” using python with the command:
    a. Python classify_video.py --input <video’s file name>
    b. Example: “python classify_video.py --input video_1.mp4”
    *c. *Note: not full path, just the video file’s name.*
3. Sit back and observe the output as below:
4. 

5. Press “Q” to exit the program using (keyboard)

Conclusion:

Thank you for taking your time in evaluating my trained model with the methods as above. I would like to confess that my model might not be very high in accuracy since I have limited resources in training process. However, I will be improving the model from current time being and please let me know if you have any doubts or questions regarding to my implementation and model.

Questions email at: jasonakon@gmail.com