UDACITY

‹ Return to "Deep Learning" in the classroom

Dog Breed Classifier

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

😃 Remarkable Work. You have acquired all the important concepts from this project.
Here are some resources for you to learn more about CNN and Transfer Learning:

- Transfer Learning and Fine-tuning
- Transfer Learning using Keras
- CS231n: Convolutional Neural Networks for Visual Recognition
- Building an Image Classifier
- Do Better ImageNet Models Transfer Better?
- Predict Age and Gender using Convolutional Neural Network and openCV

A list of best deep learning books to help you in your further study

Another list of Hands-on books on AI-ML-DL by an Udacity Alumni which featured in Udacity_EdTalk

Files Submitted

The submission includes all required files.

Step 1: Detect Humans

The submission returns the percentage of the first 100 images in the dog and human face datasets with a detected human face.

Very good, you rightly detect the percentage of human faces in the human face and dog dataset.

You can use **List Comprehension** to refactor the code.

```
human_count = np.mean([face_detector(human_img) for human_img in human_files_short])
*100
```

The submission opines whether Haar cascades for face detection are an appropriate technique for human detection.

Your view is right, though most of the time we expect that our 'intelligent interface' should be at least able to handle small occlusions and invariance in rotation, light etc. Haar cascade in that respect is not a very good choice. HOG detectors, and deep learning models perform better in this case.

Step 2: Detect Dogs

The submission returns the percentage of the first 100 images in the dog and human face datasets with a detected dog.

Step 3: Create a CNN to Classify Dog Breeds (from Scratch)

The submission specifies a CNN architecture.

The submission specifies the number of epochs used to train the algorithm.

You can also explore Keras Early Stopping it will automatically stop once the validation accuracy/loss stops improving.

The trained model attains at least 1% accuracy on the test set.

Step 5: Create a CNN to Classify Dog Breeds

The submission downloads the bottleneck features corresponding to one of the Keras pre-trained models (VGG-19, ResNet-50, Inception, or Xception).

The submission specifies a model architecture.

The submission details why the chosen architecture succeeded in the classification task and why earlier attempts were not as successful.

The submission compiles the architecture by specifying the loss function and optimizer.

The submission uses model checkpointing to train the model and saves the model weights with the best validation loss.

The submission loads the model weights that attained the least validation loss.

Accuracy on the test set is 60% or greater.

The submission includes a function that takes a file path to an image as input and returns the dog breed that is predicted by the CNN.

Your function returns along with the predicted breed name a part of file path. You can correct it by properly slicing the string in the following command in cell 1:
`dog_names = [item[20:-1] for item in sorted(glob("/data/dog_images/train/*/"))]`
Alternatively, you can slice the string, within the function using `split` around `.`

Step 6: Write Your Algorithm

The submission uses the CNN from Step 5 to detect dog breed. The submission has different output for each detected image type (dog, human, other) and provides either predicted actual (or resembling) dog breed.

Step 7: Test Your Algorithm

The submission tests at least 6 images, including at least two human and two dog images.

Fabulous job testing your algorithm, and suggesting points for improvement.
Besides these directions, I would recommend you to look at "how to fine tune a pretrained model" from this Keras post. Another interesting resource is this post
You can also refer this research paper for solving a breed problem with GAN and data augmentation, hope it can bring you a different perspective for solving this problem。

DOWNLOAD PROJECT

RETURN TO PATH