UDACITY

‹ Return to "Deep Learning" in the classroom

DISCUSS ON STUDENT HUB

Dog Breed Classifier

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Great job, you are ready to go! 💯Clearly, you have acquired all the important concepts from this project. Wish you all the best for the upcoming projects! 🤞Regarding the Web API, I would suggest that you can try to deploy your model with SageMaker.

For further improving your skills, I encourage you to take part in one of the playground competitions on Kaggle like Dogs vs. Cats Redux: Kernels Edition and Leaf Classification. As soon as you start to participate in Kaggle and learn from other, you will definitely start to pick up new concepts and become an expert!

Files Submitted

The submission includes all required, complete notebook files.

Step 1: Detect Humans

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

Step 2: Detect Dogs

Use a pre-trained VGG16 Net to find the predicted class for a given image. Use this to complete a `dog_detector` function below that returns True if a dog is detected in an image (and False if not).

Tip: For `dog_detector`, you can use this way to refactor it.

```
def dog_detector(img_path):
    return 151 <= VGG16_predict(img_path) <= 268
```

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

I would suggest that you could use the same function to tackle this problem and question 1.👇

```
def get_detector_count(detector, data_files):
    return sum([detector(f) for f in data_files])
```

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

Write three separate data loaders for the training, validation, and test datasets of dog images. These images should be pre-processed to be of the correct size.

Tip: I strongly suggest that you should read this discussion to better understand why we separate our data sets into three parts.

Answer describes how the images were pre-processed and/or augmented.

Tip: I encourage you to read this article to learn why we should apply data augmentation.

The submission specifies a CNN architecture.

Tip: I recommend you to read this article to learn the tricks to increase the performance, especially the section of activation functions. Also, I would suggest that you may consider using `nn.BatchNorm2d` to speed up the learning process.

Answer describes the reasoning behind the selection of layer types.

Choose appropriate loss and optimization functions for this classification task. Train the model for a number of epochs and save the "best" result.

Tip: Now, you successfully train your model. I suggest that you could read this discussion to learn how to analyze if your model is over-/under-fitting. Also, please look at this great discussion to get some ideas about how to handle over-fitting.

The trained model attains at least 10% accuracy on the test set.

Step 4: Create a CNN Using Transfer Learning

The submission specifies a model architecture that uses part of a pre-trained model.

Great choice, you correctly build a pre-trained model with vgg16.💯

The submission details why the chosen architecture is suitable for this classification task.

You clearly explain why you choose this model! 🎉
Agreed, we should just keep the model simple and leverage the power of transfer learning with the general features extracted from a pre-trained model! Please look at this gentle introduction of how this works.

Train your model for a number of epochs and save the result wth the lowest 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.

Step 5: Write Your Algorithm

The submission uses the CNN from the previous step 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 6: Test Your Algorithm

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

Submission provides at least three possible points of improvement for the classification algorithm.

Tip: Besides the directions you have pointed out, I would like to suggest that you may consider different models to solve the problem of face detection, please look at this great tutorial. Here is a 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