UDACITY

< Return to "Deep Learning" in the classroom

Dog Breed Classifier

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

This is one of the best submission i have seen. I liked your approach and way to tackle the problem. You thought about all teh aspects and it clearly shows your thoroughness on the subject and your want to leant more.
I gave a couple of suggestions try them and see if you get even better in your scratch model.
All the best for future projects

Files Submitted

The submission includes all required, complete notebook files.

all the necessary files present 👍

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.

Good. The percentage of the first 100 images in human_filenames and in dog_filenames were detected and correctly returned. Ideally, we would like to have 100% of human images with a detected face and 0% of dog images with a detected face but the algorithm falls short as shown in the answers with 98% of human and 17% of dog images.

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).

you rightly leveraged vgg16
I liked that you put an extra effort to identify some image take time to load or not even load so set LOAD_TRUNCATED_IMAGES=True

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

great work. the channel based normalization seems more close to ImageNet

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.

good job with implementing 3 loaders correctly

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

you could leverage more augmentation technique like Center Crop, Five Crops (4 corners + center)

The submission specifies a CNN architecture.

apart for augmentation you can try different weight initializers - kaiming initialization. I have seen people getting tremendous boost in accuracy by leveraging just kaiming.
Also, ELU activation is gaining in popularity vs Relu though they are similar.
https://pytorch.org/docs/master/nn.html#torch.nn.ELU

Answer describes the reasoning behind the selection of layer types.

good explanation

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

good job with trying Adam

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

37% is awesome. There is a limit to which we could go its purely because of the limitation on the dataset. We dont have enough data so try to find a network that could give you too high an accuracy is not possible. Still I have seen 3-4 case where people reach near 40% (like you) and only a couple where people could achieve near 50%.
You have done a great job to achieve such a high accuracy

Step 4: Create a CNN Using Transfer Learning

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

good job with leveraging ResNet. It gives the highest accuracy.

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

good explanation. ResNet as trained on a lot of dog images hence it is able to identify the difference.

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.

90% is the highest that I have seen so far. The maximum prior to you was 89%. So really have done a great job. It is worth an applaud

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.

great blend of images. I like the way you presented the actual images too for comparison

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

awesome points

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review