UDACITY

< Return to Classroom

Dog Breed Classifier

REVIEW

HISTORY

Meets Specifications

Awesome submission ! 🎉 You've done a great job of implementing a Convolutional Neural Network. Your understanding and clarity of concepts of how CNNs work is vividly depicted here.

If you are curious/want to learn more, I recommend you to take a look at the links mentioned below:
CNN's for Visual Recognition
Building an Image Classifier
Large Scale Image Recognition using CNN's
Transfer Learning
CNN Tricks
Keep learning ! ✌🏽

Files Submitted

The submission includes all required files.

All the necessary files are included 👍🏽

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.

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

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.

Your values for true positives and false positives are acceptable. Superb work in getting the correct percent of dogs and humans!

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

The submission specifies a CNN architecture.

Good job discussing the architecture of your CNN.

- Pooling is a great way to reduce network parameters. It also introduces invariance in the model. You can read about uses of pooling and different pooling layers here https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/
- Tip: Over the years, the research community has also tried an all convolutional network. Suggested paper https://arxiv.org/abs/1412.6806
- Dropout is an incredible technique to prevent overfitting of networks. Tip: Use dropout = 0.5 as that's what works well for everyone unless you have an experimental analysis or an intuitive reason for taking some other value.
- Batch normalization can be used in CNNs to help deal with internal covariate shift. A useful article https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c
- Using ReLu is a good idea. The post here details some other activation functions used in neural networks http://cs231n.github.io/neural-networks-1/
- This stackexchange question has some ideas on how to improve for architecture for better accuracy.

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

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.

Thank you for providing proper reasoning for the network's layers.

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

Good work on using `crossentropy` loss and `adam` optimizer.
Further reading: Overview of Gradient descent optimizers

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.

Superb work!

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 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.

Good set of sample images, some humans and dogs to test the algorithm.

Includes good ideas for next direction to take this for improvement.
A resource if you care to read about other general improvements.
https://machinelearningmastery.com/improve-deep-learning-performance/

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START