UDACITY

Return to Classroom

Dog Breed Classifier

REVIEW

HISTORY

Meets Specifications

Good work on the submission. I suggest you to take the same model and implement improvements. Maybe even create an application out of it. All the best!

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.

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

Very insightful answer!
Your opinion about the demand for the algorithm to provide clear images being unreasonable is great.
The technical insight about how pooling layers can still capture important features even from an unclear image is commendable.

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.

That is good architecture and you met the requirements for the project.
You can further improve the architecture with some small changes:

1. The model is overfitting, data-augmentation and normalization will take this to the next level.
2. Although Dense layer is needed for the model to be more precise it comes at the cost of huge parameter size, as you see in the summary. Try using just one dense layer, the last layer. Replace the first one with GlobalAveragePooling2D.
3. Try using different kernel sizes, especially 3x3 and see what effect it has on the model.

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

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

Test accuracy: 9.2105% in 20 epochs is a good score.

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.

You chose InceptionV3_model, can you think why this is a good choice?
Also you mentioned "Create a max pooling layer at the end of original network to decrease the spatial

dimentions". I suppose you were actually referring to GlobalAveragePooling2D, which you have actually used in the code.

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

Good use of standard 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.

Good job with the Test accuracy: 77.3923%.

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.

Precise implementation.

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.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review