UDACITY

Return to Classroom

Dog Breed Classifier

REVIEW

HISTORY

Requires Changes

3 specifications require changes

This is a good first attempt the project! There are just a few changes to be made in the notebook. See the comments below for details.

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.

AWESOME

The percentage of detected human faces in the sample datasets is correct!

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

REQUIRED

There should be a response to question 2.

Question 2: This algorithmic choice necessitates that we communicate to the user that we accept human images only when they provide a clear view of a face (otherwise, we risk having unnecessarily frustrated users!). In your opinion, is this a reasonable expectation to pose on the user? If not, can you think of a way to detect humans in images that does not necessitate an image with a clearly presented face?

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.

AWESOME

The percentage of detected dogs in the sample datasets is correct!

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

The submission specifies a CNN architecture.

REQUIRED

There should be a response to question 4:

Question 4: Outline the steps you took to get to your final CNN architecture and your reasoning at each step. If you chose to use the hinted architecture above, describe why you think that CNN architecture should work well for the image classification task.

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

SUGGESTION

It seems as though there is room for an improvement in accuracy. Try to increase the number of epochs to see if this is true. Also, to not worry about manually tuning the number of epochs, I recommend using an early stopping callback that Keras provides. Early stopping will automatically end training based on some specified rule, such as the accuracy not improving after a certain number of epochs. Take a look at Keras' callback documentation for more information.

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

AWESOME

The test accuracy is above 1%. Good job!

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.

REQUIRED

There should be a response to question 5:

Question 5: Outline the steps you took to get to your final CNN architecture and your reasoning at each step. Describe why you think the architecture is suitable for the current problem.

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.

SUGGESTION

To reduce the length of the output during training, set the `verbose` parameter in the model checkpoint to zero, like this:

```
checkpointer = ModelCheckpoint(filepath = 'saved_models/weights.best.Xception.hdf5',
                               verbose = 0,
                               save_best_only = True)
```

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.

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.

AWESOME

The function has a different output for each detected image type (dog, human, other). Nice work!

Step 7: Test Your Algorithm

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

RESUBMIT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH