‹ Return to "Deep Learning" in the classroom

Dog Breed Classifier

REVIEW | HISTORY

Meets Specifications

Dear Learner,
I can see you did read comments from the previous review and followed the required guidelines to adjust your answer carefully. Excellent work!!!👍
And I'm really glad that you paid so much patience to read the feedback and followed them to carefully amend the answers. You have a really great learning attitude, keep it UP!💪
The work now meets all of the specifications. Congratulation. You made it!👏

Files Submitted

The submission includes all required files.
All necessary files are present in this submission. Great!

Rate this review

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.

Superb work in getting the number of images in the first 100 images of dogs and humans faces datasets with detected human faces!

```
The detector has found 100.0% of the images in human_files_short to include human faces
The detector has found 11.0% of the images in dog_files_short to include human faces
```

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

Great job!!

> I think this is a fair expectation to pose on the user to ensure we get good results. However, in a world of competing algorithms, other algorithms may not have this expectation and thus can be more attractive for people to use. Therefore, to ensure that our algorithm does well relative to competition, it would be useful to detect humans in images without requiring a clearly presented face. Apart from OpenCV, the dlib library offers useful face detector algorithms.

That is good intuition and I have to agree with you. I think there should be more algorithms that could detect more obscure faces from different angles. It would be a lot more complex but it would make most image classifiers' functionalities a lot better.

Pro Tips

Here are some documents that provide more information on Haar cascades.

- Tutorial face detection;
- https://www.youtube.com/watch?v=88HdqNDQsEk;
- Opencv face detection;
- OBJECT DETECTION : FACE DETECTION USING HAAR CASCADE CLASSFIERS;
- Youtube video;

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.

Superb work in getting the number of images in the first 100 images of dogs and humans faces datasets with a detected dog!

```
The detector has found 0.0% of the images in human_files_short to include dogs
The detector has found 100.0% of the images in dog_files_short to include dogs
```

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

The submission specifies a CNN architecture.

The submission perfectly specifies the architecture used. Well done!👍

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

We note the number of epochs (`epochs = 20`) in this submission.

Pro Tips

Note that, during training, if the training loss decreases but the validation loss does not, then that model may be overfitting the training data. The model with the lowest validation loss is typically best.
Here are some documents that talk about the choice of the number of epochs.

- How does one choose optimal number of epochs?
- How to train your Deep Neural Network;
- Number of epochs to train on.

Rate this review

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

A good result of your model on the test dataset of dog images.

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 correctly downloads the bottleneck features. Nice work!👍

Suggestions and Comments

- ResNet50 is a popular choice and here is a nice article which discusses it further. You may opt to use it again someday since it is one of the most popular pre-trained models from Keras.
- Also, try using the other bottleneck features and compare the results obtained. **Xception**, for example, outperforms the others. Find out why

The submission specifies a model architecture.

Good work! The submission specifies a model architecture. Very good implementation of transfer learning!👍

The submission details why the chosen architecture succeeded in the classification task and why earlier attempts were not as successful.

Nice work! This submission answers question 5.

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

The `categorical_crossentropy` loss function and `rmsprop` optimizer are specified in the compilation of this architecture.

Pro Tips

- The Adam optimizer does sometimes work well but also can generalize poorly. Check out this article for more details.
- It's usually best to try an ada-based method (adam, nadam, etc) and SGD with momentum, and use whichever has better generalization (test/validation set) performance.
- Here is a comparison of some different optimizers.

The submission uses model checkpointing to train the model and saves the model weights with the best validation loss.

Good job! 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.

Good job loading the model weights that attained the least validation loss!

Accuracy on the test set is 60% or greater.

Superb work! Accuracy on the test set is `77.8708%`.👏🏽

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.

The submission contains 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.

Wow! The submission uses CNN from Step 5 to detect dog breed. Furthermore, the submission has different output for each detected image type (dog, human), 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.
Splendid work with testing your algorithm and getting great results. You did well on detecting and classifying dogs from humans! Great job.

DOWNLOAD PROJECT

RETURN TO PATH