UDACITY

Return to "Deep Learning" in the classroom

Dog Breed Classifier

REVIEW

CODE REVIEW

HISTORY

Requires Changes

2 SPECIFICATIONS REQUIRE CHANGES

Dear Excellent Student,
Excellent work in this project! **You have made a remarkable model and you possess great potential and have sufficient knowledge about the subject**. It is amazing you got this far with just the first shot. There are just a few minor issues to be addressed in this work and the outcome will be superb! I wish you keep studying hard and maintain your brilliance in creating wonderful models and algorithms. Have fun and enjoy learning!

> The step 5. I've tried very hard to improve performance but after getting 80% I stucked. I just wanted to get tips where my architecture can be changed in order to get even better results.

I will suggest you try out other power models like Xception for example. Some models can not be tuned beyond their performance. You neeed to try other models too to see how they perform and compare to yours. 100 epochs is too much and as you said the model started overfitting. Please try other architectures and see how they varry.

Files Submitted

The submission includes all required files.

All necessary files are present in this submission. Great!

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.

Nice work in using the function `face_detector` and getting the appropriate percentage of the files!

```
Performance in the human_files_short: 100.00%
Performance in the dog_files_short: 11.00%
```

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

> Answer: Yes I think it is possible to detect humans in images even if there is no face presented. There are other attributes in the humans bodies that I believe CNN can detect and identify to produce high performance detectors. Certainly it wouldn't be the best model but still a practicle one. We humans are able to perform such a task. We can easily detect humans and other animals looking only to their bodies. As NN mimic the way we learn to perform common tasks I do believe there is a way to train a model to do the same.

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

- Face Detection Tutorial
- Haar Cascade Object Detection Face & Eye
- OpenCV Face Detection
- Face Detection using Haar Cascade Classifiers

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.

You have achieved superb results. Your algorithm is optimal and does the job expected. Keep it up!

```
Performance in the human_files_short: 0.00%
Performance in the dog_files_short: 100.00%
```

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

The submission specifies a CNN architecture.

Superb work in describing the architecture used. It is a good idea to not stray away too much from what you have learned and use a bit of it as a base to learn and formulate better architectures. Nice job on this one.

Suggestions and Comments

You could experiment too with batch normalization in the network. This works by normalizing the weighted inputs in the network. Without it, only the inputs at the beginning of the network are normalized. More information about this found on Quora can be seen here why does batch normalization help.

- I suggest you read this document about Using Convolutional Neural Networks to Classify Dog Breeds. This is a good outside resource to complete your reading of this course.

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

Nice choice with `10` epochs in training the model. To get everything out of your network and data, the epochs should be chosen such that the validation accuracy is no longer increasing. You can do this manually or, better, use the early stopping callback function of Keras. Check this article for more information.

Pro Tips

Here are several 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.

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.

```
Test accuracy: 4.6651%
```

Suggestions and Comments

I would like to share this article with you wherein the discussion is improving the accuracy when using a CNN. This may help you out in the future.

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!👍🏽

The submission specifies a model architecture.

Fantastic architecture for your model. It would really be nice if your model outperforms our base model in this project!

Pro Tips

Please, I suggest you take a look at this very good document:

- Dog breed image classification with Keras.

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

Well done outlining the steps you took to arrive at your final architecture, however, question 5 has not been answered in full. Please read my comment below.

Required

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

What we are really interested in here is for you to give a detailed explanation of why you think the chosen architecture succeeded for the current problem and why earlier attempts were not as successful. This is not present in the answer you provided for question 5 and I think its because it was not specified in question 5, however, you need to include it to your answer. Please provide this next time to complete this section together with all the steps you took to arrive at your final architecture.

Suggestions and Comments

Even though the architecture chosen worked out as expected, it would be good to try out or make other attempts with different architectures and find out the reason they aren't working the way they are supposed to be or why they work better. Try doing this next time. I believe it will enhance your knowledge and skills in making better decisions and architectures in the future.

Xception, for example, gives better results. You could find out why.

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

The loss function and optimizer are specified in the compilation of this architecture. That's great!

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.

Impressive work in calculating for the accuracy! The accuracy on the test set easily achieved `80.6220%`, great.

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 the CNN from Step 5 to detect dog breed. Furthermore, 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.

Fantastic work with utilizing your function in classifying the images and testing it on both humans and dogs. However, you may have overlooked a question here. 😃

> Question 6: Is the output better than you expected :) ? Or worse :(? Provide at least three possible points of improvement for your algorithm.

Required

- Please provide a brief discussion on your output and provide at least three possible points of improvement for your algorithm.

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)