Back to Deep Learning Nanodegree

Dog Breed Classifier

REVIEW | CODE REVIEW | HISTORY

Meets Specifications

Keen Student,
Superb submission! You have good potential and well-versed on our subject. Its quite great that such brilliant performance was obtained after carefully following the previous review to amend your points. Excellent work in keeping your answers concise and straightforward! Keep up the fantastic job and I hope you do well and have fun studying the course!👍

The Accuracy you obtained(82.0574%) in this work is superb and outshines our benchmark accuracy of 60%. Such great value is rarely attained by most of our students. You deserve some praises for this👏👏

```
This great value could earn you a nomination for this work. Please sacrifice a minute or two of your time to provide me
```

Were they any major difficulties encountered? What was the most amazing and interesting part of this project? please let me know after you rate this work 🙏

```
I await your kind response, thanks 🙏
```

Files Submitted

	The submission includes all required files.
	Great job compiling the necessary 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.
	Fantastic job in getting the correct percentage of humans and dogs! 😃

	The submission opines whether Haar cascades for face detection are an appropriate technique for human detection.
	Well done! The response provided is logical and clearly expresses your opinion. **Tips** Here are some documents that talk about Haar cascades, take an inspiration from this and complete your answer using Haar cascades. • Object detection using Haar-cascade Classifier • OBJECT DETECTION : FACE DETECTION USING HAAR CASCADE CLASSFIERS

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.
	Good job on this one!🎉

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

The submission specifies a CNN architecture.

This is remarkable work in specifying the architecture. You have good knowledge in model architecture and choosing only the ideal steps for optimal output.👍🏽

Comments

- Take a look at 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.

An epoch number of 3 is interesting.

Comments

- Do you think a number greater than 10 would be better in assessing the accuracy?
- I highly suggest next time for you to try a bigger number of epoch, like 15. It may take longer but it would provide you better accuracy.
- Considering the speed and the capacity of the processor, a lower number is also acceptable.

Tips

Here are some documents that will better explain the choice of the number of epochs to use;

- 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.

Wow, `1.0766%` is an excellent number for the test set and beats the expected 1%👏🏽

Comments

In this article, discussion on how to improve the accuracy using CNN is given. This may help you out in the future, do take a look at it😃

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 bottleneck features are downloaded. Good job.

The submission specifies a model architecture.

Great architecture for your model. I hope your model outshines the base model and it seems that would be the outcome by the way it looks.

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

Remarkable work in finding ways to improve the model and finding a way to garner more impressive output from the model.

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

Good use of the loss function and optimizer!

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

Impressive implementation of the model checkpointing!

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.

A remarkable accuracy of 82.0574%! Impressive work so far!👏👏

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.

Great work in implementing a function that takes the file path to an image as an input. This would make the function re-usable and would make the model's job easier as it just needs to recycle the function with a different path per image.🎉

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.

The implementation is great and uses the CNN from the previous step to detect the dog breeds! It would be fun to predict what kind of dog breed some humans are using your model! 😃

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 superb results. You did a fine job in detecting and classifying the dog breeds and humans alike! 😃

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ