UDACITY

Back to Deep Learning Nanodegree

Dog Breed Classifier

审阅

HISTORY

Meets Specifications

Dear Excellent Student,
Thank you for this submission. It is great and meets all of our expectation. A lot of work has been done in this project and you should be proud of yourself. I will suggest you to implement the suggestions you mentioned. This will enhance your knowledge on the domain. Keep up the good work and continue working hard in other projects.

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.

- The percent of correct human identification: 100%.
- The percent of dogs misclassified as humans: 11%.
  The above values are acceptable. Good job!

Superb work in getting the correct percentage of dogs and humans!

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

That is good intuition and I have to agree with you. I think there should be more algorithms that could detect more obscure faces on 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 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;
- Adam Harvey Explains Viola-Jones Face Detection.

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.

Percent humans misclassified as dogs: 0%.
Percent of dogs correctly classified: 100%.

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

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

The submission specifies a CNN architecture.

The submission perfectly describes the architecture used.

Suggestions and Comments

- 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.

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: 2.2727%
```

Suggestions and Comments

I would like to share this article with you wherein the discussion is improving the accuracy using 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 models 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.

Excellent work! It is good to know that the chosen architecture does a better job than the original classifier we had. It is essential to distinguish dog faces from human faces and your architecture seems to be a good fit.

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.

Nice work with 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.

Fantastic implementation in calculating for the accuracy!

- VGG19 Test accuracy: 55.8612%
- ResNet50 Test accuracy: 82.8947%
- Inception Test accuracy: 86.4833%
- Xception Test accuracy: 81.9378%

We see that Inception performs the best amongst all with a very high accuracy.

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.

Excellent work with testing your algorithm and getting great results. You did well on detecting and classifying the dog breeds. It is still interesting on how the model classifies humans as dogs based on its algorithm.

下载项目

返回 PATH