Meets Specifications

Brilliant Learner,  
We thank you for the efforts made in this project. The submission is good and I must admit that you have showcased a deep understanding of this project. I'm very confident to validate this project and you should be proud of your work. Keep up with this working spirit and other submissions will always be awesome. Congratulations!  
The following links may be of great interest:

* [How to improve my test accuracy using CNN in Tensorflow](https://datascience.stackexchange.com/questions/20104/how-to-improve-my-test-accuracy-using-cnn-in-tensorflow).
* [How To Improve Deep Learning Performance](https://machinelearningmastery.com/improve-deep-learning-performance/).
* [A Guide to TF Layers: Building a Convolutional Neural Network](https://www.tensorflow.org/tutorials/layers).

**Files Submitted**

**The submission includes all required files.**

Thank you for submitting all the necessary files required for a complete review of the project.

**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.**

Excellent job in retrieving the right percentage of humans and dogs.

Dog Detection percentage -> 11.00%  
Human Detection percentage -> 100.00%

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

Logical thinking!.

**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.**

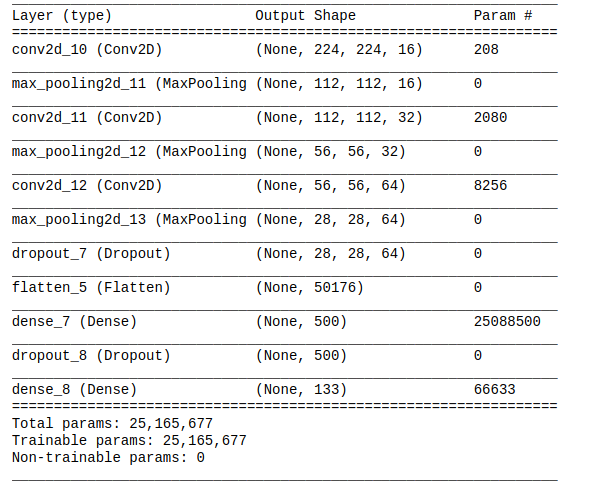
Great job! You have successfully retrieved the correct percentages of the images!

Human Detection percentage -> 100.00%  
Dog Detection percentage -> 0.00%

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

**The submission specifies a CNN architecture.**

A wonderful job was done in creating a CNN in this section of the project as required. The architecture of the CNN was also provided in the notebook. Great job!

[](https://udacity-reviews-uploads.s3.us-west-2.amazonaws.com/_attachments/38745/1525924515/Selection_172.png)

**Admonition**

Please, the following links may be of great interest to you for further research:

* [Keras Tutorial: The Ultimate Beginner’s Guide to Deep Learning in Python](https://elitedatascience.com/keras-tutorial-deep-learning-in-python).
* [Deep learning for complete beginners: convolutional neural networks with keras](https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html).
* [keras tutorial – build a convolutional neural network in 11 lines](http://adventuresinmachinelearning.com/keras-tutorial-cnn-11-lines/).
* [Image Classification using Convolutional Neural Networks in Keras](https://www.learnopencv.com/image-classification-using-convolutional-neural-networks-in-keras/).

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

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

**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.**

A nice job was done in implementing Resnet and Inception models, testing them and then choosing Resnet with the best accuracy. This was also well documented in the notebook. Keep it up!

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

The solution was compiled with the optimizer and loss function specified

**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.**

**Accuracy on the test set is 60% or greater.**

81.4593% is a superb output! Keep up the great work!

**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.**

A function rasnet50\_predict\_dogbreed(img\_path) was correctly defined to take a file path as input and return the breed predicted by the CNN in the solution. Great job!

**Step 6: Write Your Algorithm**

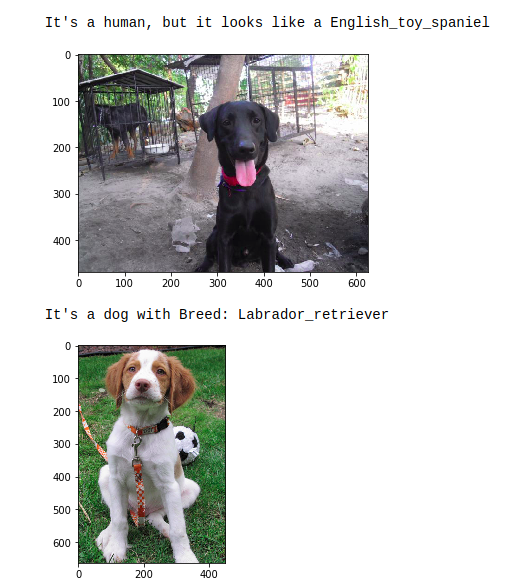
**The submission uses the CNN from Step 5 to detect dog breed. The submission has different output for each detected image type (dog, human, other) and provides either predicted actual (or resembling) dog breed.**

Great work done here.

**Step 7: Test Your Algorithm**

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

I am glad to see that you have successfully tested the images on dogs and humans alike! It is good to know that your model does a great job in classifying the images properly and you have provided a brief discussion on how to improve the model.

[](https://udacity-reviews-uploads.s3.us-west-2.amazonaws.com/_attachments/38745/1525925401/Selection_173.png)