* Logout

**PROJECT**

Dog Breed Classifier

A part of the Deep Learning Nanodegree Program

* **PROJECT REVIEW**
* **NOTES**

**SHARE YOUR ACCOMPLISHMENT**

Requires Changes

**2 SPECIFICATIONS REQUIRE CHANGES**

Great job, you are almost there! 💯Clearly, you have acquired all the important concepts from this project. You only need to make some modification and then you are ready to go. Wish you all the best for the upcoming projects!🤞

I think that you are ready to take part in one of the playground competitions on Kaggle like [this one](https://www.kaggle.com/c/dog-breed-identification) and [this one](https://www.kaggle.com/c/plant-seedlings-classification). Please check out the kernels to learn techniques and tricks from other people and see if you can get a top leaderboard score!

**Files Submitted**

**The submission includes all required 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.**

Clean code! 👏  
Tip: I would suggest that you could use **List Comprehension** and **format** to refactor your code.

pred\_human\_percentage = np.mean([face\_detector(f\_human\_path) for f\_human\_path in human\_files\_short])

print("Human Detection percentage -> {:.2%}".format(pred\_human\_sum))

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

Agreed, CNN could be a possible solution. 👍  
In my personal opinion, I think that it depends on the scenarios. For identity identification, it’s reasonable to ask users to submit a clear image. But for large-scale images searching, it may be not reasonable to ask for a clear image.  
Tip: The only thing I want to add is that Haar-like features are rotation invariant, which means that the features will not work when the image is rotated. Please check [this discussion](https://www.quora.com/How-can-I-understand-Haar-like-feature-for-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.**

I would suggest that you could use the same function to tackle this problem and question 1.👇

def get\_detector\_percentage(detector, data\_files):

return np.mean([detector(f) for f in data\_files])

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

**The submission specifies a CNN architecture.**

Great details about your logic behind this model! 💯it definitely shows that you have a strong intuition about building a model.  
Tip: I want to point out that when the kernel size is small, it can capture more fine detail, such as the hair from a cat image. You can check [this great article](https://hackernoon.com/visualizing-parts-of-convolutional-neural-networks-using-keras-and-cats-5cc01b214e59). Also, I would suggest that you could use **Batch Normalization** to see if it can help retain higher performance, please check [this great video](https://www.coursera.org/learn/deep-neural-network/lecture/81oTm/why-does-batch-norm-work).

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

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

Outstanding! 👍

**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.**

Great model! 💯We should just keep the model simple and leverage the power of transfer learning! But please provide your reasoning about why transfer learning could outperform the scratch model, such as why transfer learning needs a pre-trained model.  
Tip: I suggest that you could analyze whether your model is overfitting, the systematic way is plotting the training accuracy and testing accuracy along with epochs, please check [this great tutorial](https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/).

Question 5: Outline the steps you took to get to your final CNN architecture and your reasoning at each step. Describe why you think the architecture is suitable for the current problem.  
**The submission details why the chosen architecture succeeded in the classification task and why earlier attempts were not as successful.**

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

**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.**

You outperform the baseline! 🎉

**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.**

**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.**

**Step 7: Test Your Algorithm**

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

Great testing images! 🎉Please use at least two human images as the testing images. Also, we would like to see your thoughts on how to improve your current algorithm. Please kindly share your ideas.  
Tip: I would like to suggest that you may consider different models to solve the problem of face detection, please look at [this great tutorial](https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78). Here is [a research paper](https://www.ams.giti.waseda.ac.jp/data/pdf-files/2017_IEVC_watabe.pdf) for solving a breed problem with [GAN](https://julianzaidi.wordpress.com/2017/04/24/deep-convolution-gan-dcgan-architecture-and-training/) and data augmentation, hope it can bring you a different perspective for solving this problem。

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

 RESUBMIT PROJECT

[**DOWNLOAD PROJECT**](https://review-api.udacity.com/api/v1/submissions/1204499/archive)



**Best practices for your project resubmission**

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

 Watch Video (3:01)

RETURN TO PATH

Rate this review

( )( )( )( )( )

* [**Student FAQ**](https://review.udacity.com/#!/submissions/student-faq)