Summary

This package contains the solution to the dog-breed classification project, please refer to the modified notebook, dog_app.ipynb, and the inline answers to the questions.

The project was completed using Colab Pro, so there are some additional cells dealing with copying the data from drive and downloading the images.

Please refer to the answers on the notebook for the chosen network architectures.

The network trained from scratch obtained an accuracy of **18%,** whereas the one with transfer learning (based on VGG19) obtained an accuracy of **89%**.

Improvements

I suspect that parameter initialization, learning rate and optimizer in the network from scratch could be optimized further. I had several attempts but in the end only a vanilla SGD with learning rate of 0.001 would reliably make the loss decrease, and so I continued with this approach. The loss did decrease, and could have continued decreasing even beyond the 100 epochs training, albeit very slowly.

The dropout was quite high, I got this from the VGG paper, but it's possible that because I added batch normalization it could be reduced. I did however observe significant overfitting with dropout probability of 0.1. So I increased it to 0.5 and tha solved overfitting, but perhaps a lower value would work better.

The face and dog detectors functions could use some improvement, perhaps by replacing VGG16 with VGG19 or by using a different OpenCV face detector. In the final exercise, one human and one cat were reported as dogs, and a mutt dog was not recognised as a dog.

In order to improve the accuracy of the transfer learning network, I think longer training should be advantageous since the cost was still decreasing after 100 epochs. Further experimentation with batch normalization and different dropout could also be useful. Lastly, again, it's possible that using a different optimiser and learning rate might also speed up training.