**Weekly report**

Set out three main aims for the next few weeks:

1) Implement a basic form of transfer learning, perhaps using models from Image net

2) Implement TFRecords, the standard data input form in Tensorflow

3) Expand our possible input of characters (and increase our number of samples) by using the whole CASIA isolated dataset

Elliot will work on (1), Seb will work on (2, 3).

*Work on TFRecords and CASIA dataset*

Went back to basics and tried to increase the efficiency in the way in which we resized a character.

Realised this isn’t necessary, since we only generate the character images once.

Generated the character images again, this time to a larger size (as suggested in literature) of 48x48, including 2 pixels of padding on each side.

Incorporated 4030 unique characters, instead of 3755 as before. This includes alphanumeric characters (a,b,… 1,2,…)

TFRecords requires images are saved as images, not arrays – this was done and each image was labelled.

All .gnt files from the 1.0 dataset can now be loaded, and they are separated into training and test samples as recommended by CASIA

Wrote to TFRecords successfully. Still need to read from these and implement them in a network – and this is the only way we can check if they were written correctly.

*Work on transfer learning*

Researching into transfer learning has been quite interesting. It turns out that it comes down to the second to last layer which is called the bottleneck of a network. This bottleneck layer provides a set of unique values for any image passed through the network. This includes images that the network isn’t trained to classify.

This means if one wishes to increase the number of output classes, one can discard the final layer and replace it with an untrained layer with the desired number of outputs. This final layer can then be trained with the others kept the same. To further increase efficiency one can precompute the bottleneck values for each image in the set before training and then the problem reduces to a simple fully connected neural net which will train in half an hour on a laptop.

So far I have been able to save and load simple variables in TensorFlow. This week I will apply this to a neural network so that I have the necessary tools to perform transfer learning.

**Action points for the next week**

1. Write code to read from TFRecords, and check that it works properly as part of a neural net

2. Incorporate all 1.0 dataset .gnt files, then 1.1 and 1.2 datasets

3. Figure out how to do shuffling and image deformation with TFRecords.

4.Create a script to Load a trained neural network model and train only the last layer on some data.

5.Do some further research into the bottleneck layer and why it provides a unique representation of each piece of input data (Theory).