1. ABSTRACT

In this project a deep learning algorithm was developed using transfer learning to retrain the **Incenptionv3** convolutional neural network multiple times. The algorithm uses data from 516 Diabetic Retinopathy images (more information about the data can be found here: https://idrid.grand-challenge.org/Home/) provided for this project. The algorithm trains on 413 images and is tested on 103 images.

To train the algorithm, transfer learning is used to retrain the Inceptionv3 CNN 11 different times. A new data set is used each time the CNN is retrained. The data is divided into 5 classes(0, 1, 2, 3, 4). The first time the CNN is retrained, all the classes are used for training and testing. The remaining 10 times the CNN is retrained, only a combination of 2 classes are used for testing and training. Using 2 sets at a time, with 5 different sets of data, gives 10 possible combinations. Every time the CNN is retrained, new predicted probabilities are calculated. The predicted probabilities are all added together to create a single set of predictied probabilities. After retraining the CNN 11 times, each class for each sample will have a summation of 5 different probabilities. The maximum probability score for each class is 5.0 (minimum 0.0). From the final predicted set of probabilities, the algorithm classifies each image by choosing the class with the highest score.

The algorithm follows this order:

1. Retrain CNN using all classes of data
2. Make a matrix with 'Samples x Classes' size and save all probabilites within this matrix
3. Retrain CNN using one combination of 2 Classes
4. Add the probabilities to the matrix made in step 2
5. Start at step 3 again (go to step 6 only when all combinations of 2 classes have been used to retrain the CNN)
6. Classify all samples based on the class with the highest probability score

2. TEST RESULTS

Every time the script is runs, the results change. In my tests, the algorithm classifies the test data with greater than 90% accuracy. The algorithm never classifies images in Class 1 correctly but this class has the fewest testing and training images. See page 3 for a confusion matrix chart. See page 4-7 for the final probability matrix. All misclassified cells and their associated probability scores are highlighted in red. All labels of misclassified data and their associated probability scores are highlighted in green. Of the 7 incorrect classifications, 5 of them are of Class 1 and the remaining 2 are of Class 4 (the 2nd smallest data class). For all misclassifications, the 2nd best probability score was the correct class and within 0.3 points (roughly 6% [0.3/5.0]). Page 8 shows a bar graph representing the overall classification accuracy for each retraining of the CNN. When the CNN was trained using all the classes is when the classification accuracy was the worst.

If this algorithm was given a balanced amount of training/testing data, it appears that it would perform very well.

The main script of the algorithm is main.m