# SFWRTECH 4NN3 Neural Network Final Project

**Goh Wei Pin, S.N: 400488874**
**Lee Zi Guang, S.N: 400488873**

## Pre-processing

We used the images provided with the project requirements documents: lfw-deepfunneled, non_faces, and testing. Since the images in lfw-deepfunneled were nested in subdirectories, we wrote a simple function, **singleFolder** to move all the images into another folder "data" for easier data pre-processing.

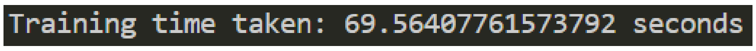
The pre-processing performed on the images consists of:

1. Converting .jpg into numpy arrays
2. Greyscaling
3. Resizing them into 32x32 pixels for easier training with a reduced input size
4. Flattening the now 2D greyscale into 1D so that it can be an input for our training

By using skimage.transform.resize, we retain the information from the original image even after resizing. We then flattened the np array

## Training the model

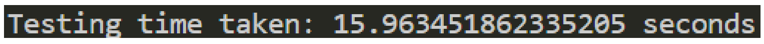
We decided to label images with faces as class 1, and images with no face as class 0. We accomplished this by creating numpy arrays of 0 and 1 based on the number of images without faces and with faces respectively. We concatenate the arrays with faces and arrays without faces into a single array, and also concatenate the labels similarly. These 2 arrays are then used to train the model using scikit-learn's **MLPClassifier**. We also timed the computational timing for both the training and testing.



```
Training time taken: 69.56407761573792 seconds
```

## Testing

We conducted testing by using a 32x32 sliding window across the images, moving from the top-left corner, making rightward shifts first, then downward shifts. We added each row into an array iteratively, making the resulting list a 1D array, so there is no need to flatten it before feeding it into our model.

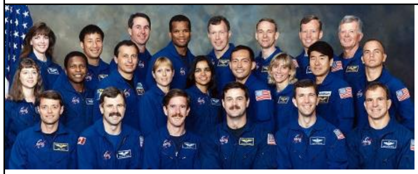
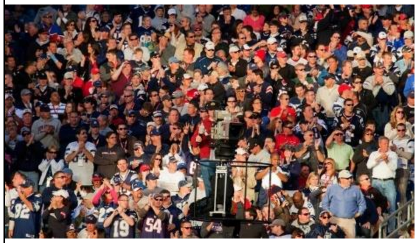
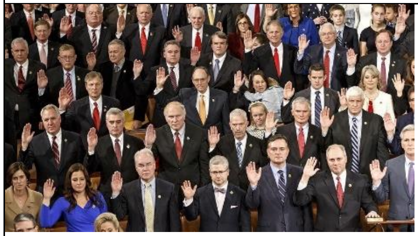
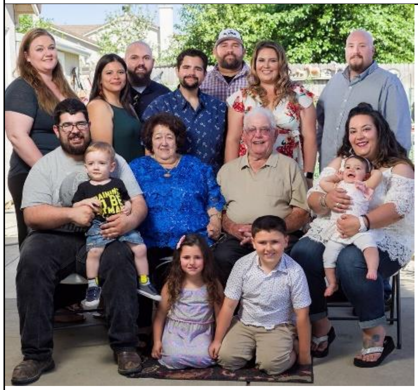
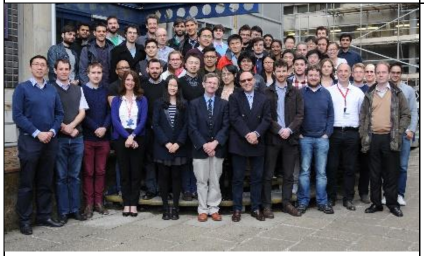
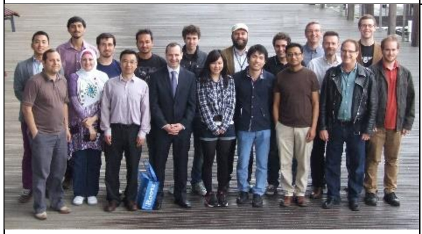


```
Testing time taken: 15.963451862335205 seconds
```

## Results

Since there was no ground truth provided, we decided to manually count the number of faces in each picture. We partitioned each test image into 32x32 blocks and checked if it contained a face. (For blocks with part of a face, if the face took >50% of the block, we counted it as a face).

As such, our initial testing methodology did not make sense, so we changed it from shifting by +1, to shifting by +32. Using our manually counted list, we compared it with our model's prediction to come up with the confusion matrix. This confusion matrix is stored as a .json file, "cf.json".

| Image | True Positive | False Positive | False Negative | True Negative |
|---|---|---|---|---|
|  | 14 | 4 | 7 | 135 |
|  | 53 | 11 | 24 | 324 |
|  | 80 | 17 | 44 | 676 |
|  | 101 | 20 | 55 | 1424 |
|  | 129 | 20 | 69 | 4498 |
|  | 293 | 22 | 81 | 12840 |

| | | | | |
|---|---|---|---|---|
| | 303 | 25 | 84 | 13124 |