Assignment 2B
Kritagya Nepal (kxn190007)
Name of Group Member: None
Number of free days used: 3

Face Expression Recognition using Convolution Neural Net

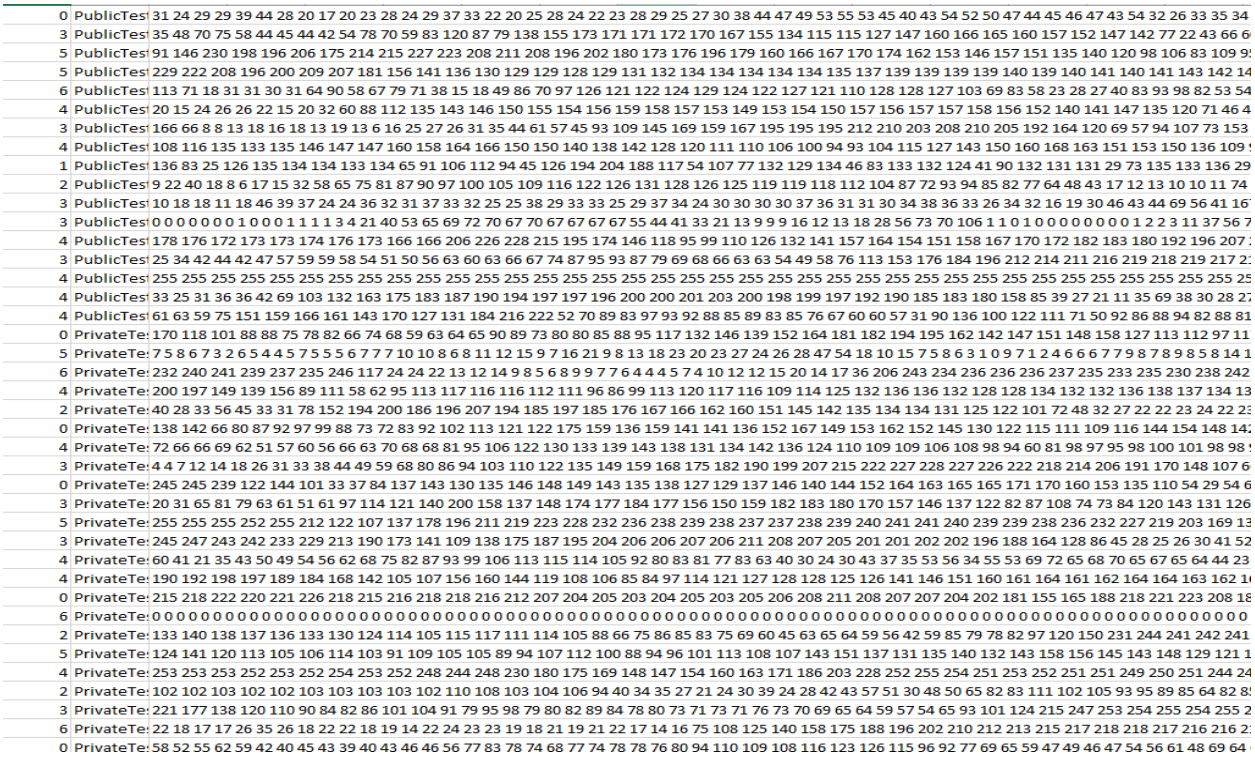
For image classification, the facial expression recognition is used. The dataset consists of seven different emotion which includes, 0 = Angry, 1 = Disgust, 2 = Fear, 3 = Happy, 4 = Sad, 5 = Surprise, 6 = Neutral. The dataset consists of 3 columns, which includes emotion, Usage and pixels respectively. There are total of 28709 training data and 7178 testing data. The pixel size for the image is 48*48 pixel. The link to the project can be found here:

Link

The training and testing data are in a csv file that looks like this:

emotion	Usage	pixels
0	Training	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 119 115 110 98 91 84 84 90 99 110 126 143 153 158 171 169 172 169 165 129 110 113 107 95 79 66
0	Training	151 150 147 155 148 133 111 140 170 174 182 154 153 164 173 178 185 185 189 187 186 193 194 185 183 186 180 173 166 161 147 133 172 151 11
2	Training	231 212 156 164 174 138 161 173 182 200 106 38 39 74 138 161 164 179 190 201 210 216 220 224 222 218 216 213 217 220 220 218 217 212 174 1
4	Training	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 19 43 52 13 26 40 59 65 12 20 63 99 98 98 111 75 62 41 73 118 140 192 186 187 188 190 190 187 182 1
6	Training	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84 115 127 137 142 151 156 155 149 153 152 157 160 162 159 145 121 83 58 48 38 21 17 7 5 25 27 24 25 1
2	Training	55 55 55 55 55 54 60 68 54 85 151 163 170 179 181 185 188 188 191 196 189 194 198 197 195 194 190 193 195 184 175 172 161 159 158 159 147 13
4	Training	20 17 19 21 25 38 42 42 46 54 56 62 63 66 82 108 118 130 139 134 132 126 113 97 126 148 157 161 155 154 154 164 189 204 194 168 180 188 214 2
3	Training	77 78 79 79 78 75 60 55 47 48 58 73 77 79 57 50 37 44 56 70 80 82 87 91 86 80 73 66 54 57 68 69 68 68 49 46 75 71 69 70 70 72 72 71 72 74 77 76 83
3	Training	85 84 90 121 101 102 133 153 153 169 177 189 195 199 205 207 209 216 221 225 221 220 218 222 223 217 220 217 211 196 188 173 170 133 117 13
2	Training	255 254 255 254 254 179 122 107 95 124 149 150 169 178 179 179 181 181 184 190 191 191 193 190 190 195 194 192 193 196 193 192 188 182 173
0	Training	30 24 21 23 25 25 49 67 84 103 120 125 130 139 140 139 148 171 178 175 176 174 180 180 178 178 182 185 183 186 186 178 180 172 175 171 155 1
6	Training	39 75 78 58 58 45 49 48 103 156 81 45 41 38 49 56 60 49 32 31 28 52 83 81 78 75 62 31 18 19 19 20 17 20 16 15 12 10 11 10 23 36 65 59 9 3 5 7 93 6
6	Training	219 213 206 202 209 217 216 215 219 218 223 230 227 227 233 235 234 236 237 238 234 226 219 212 208 201 190 183 176 161 74 15 24 22 22 22 2
6	Training	148 144 130 129 119 122 129 131 139 153 140 128 139 144 146 143 132 133 134 130 140 142 150 152 150 134 128 149 142 138 156 155 140 136 14
3	Training	4 2 13 41 56 62 67 87 95 62 65 70 80 107 127 149 153 150 165 168 177 187 176 167 152 128 130 149 149 146 130 139 139 143 134 105 78 56 36 50
5	Training	107 107 109 109 109 109 110 101 123 140 144 144 149 153 160 161 161 167 168 169 172 172 173 175 176 171 170 166 165 162 162 157 150 149 14
3	Training	14 14 18 28 27 22 21 30 42 61 77 86 88 95 100 99 101 99 98 99 99 96 101 102 96 95 94 88 78 72 65 55 40 25 20 20 42 64 74 129 133 125 144 151 153
2	Training	255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 25
6	Training	134 124 167 180 197 194 203 210 204 203 209 204 206 211 211 216 219 224 228 230 230 226 222 220 217 217 210 207 213 210 199 191 190 188 17
4	Training	219 192 179 148 208 254 192 98 121 103 145 185 83 58 114 227 225 220 203 202 168 154 157 164 182 211 164 94 122 155 176 238 240 242 192 87
4	Training	1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 7 12 23 45 38 35 14 43 27 31 24 18 20 29 18 6 2 4 2 0 1 1 1 1 0 5 13 16 16 16 15 14 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 3 7 9 27 44 4
2	Training	174 51 37 37 38 41 22 25 22 24 35 51 70 83 98 113 119 127 136 149 149 141 125 107 77 50 30 21 9 38 96 79 72 87 60 23 25 43 29 24 33 51 36 33 26
0	Training	123 125 124 142 209 226 234 236 231 232 235 223 211 196 184 181 182 186 185 193 208 211 208 201 196 192 191 192 194 201 207 216 225 225 22
0	Training	8 9 14 21 26 32 37 46 52 62 72 70 71 73 76 83 98 92 80 90 110 148 158 149 166 172 166 166 164 179 196 197 189 185 194 189 174 173 177 179 186
3	Training	252 250 246 229 182 140 98 72 53 44 67 95 95 89 89 90 90 93 94 89 88 88 83 82 81 73 74 69 69 64 58 49 46 49 35 32 27 25 24 21 13 9 11 15 15 15 21
3	Training	224 227 219 217 215 210 187 177 189 200 206 212 210 208 204 207 206 207 205 203 203 206 206 201 204 199 196 189 187 187 186 185 184 186 18
5	Training	162 200 187 180 197 198 196 192 176 152 136 114 109 117 124 133 162 182 195 200 202 203 204 200 201 202 200 200 200 200 198 195 192 189 18
0	Training	236 230 225 226 228 209 199 193 196 211 199 198 194 199 214 209 202 207 179 156 154 152 157 156 151 144 136 138 140 148 183 209 192 177 15
3	Training	210 210 210 210 211 207 147 103 68 60 47 70 124 118 119 123 124 131 136 138 146 144 147 152 157 159 162 165 165 164 163 161 159 159 159 16
5	Training	50 44 74 141 187 187 169 113 80 128 181 172 76 62 37 41 40 37 55 44 37 36 48 45 40 48 82 82 78 69 56 60 50 42 52 69 125 154 128 68 57 46 50 51 4
3	Training	234 233 228 231 234 233 236 230 236 196 112 85 100 114 129 133 132 132 147 164 180 156 161 156 151 150 137 117 139 157 138 145 137 132 114
6	Training	244 214 198 187 105 89 47 38 40 28 26 29 42 39 33 48 35 22 28 31 42 46 42 58 61 54 42 46 54 43 38 49 50 86 100 101 93 65 69 71 61 114 161 211 23
3	Training	88 116 108 66 61 52 56 74 80 70 55 57 61 68 57 44 62 69 65 61 60 61 68 55 51 65 66 71 60 79 95 94 86 113 112 99 113 108 116 118 88 72 68 25 46 1
2	Training	59 63 70 67 67 72 69 80 95 105 125 141 149 155 164 168 168 174 180 181 184 187 190 194 194 196 200 201 201 200 200 196 193 191 186 181 179 1

Testing Data:



Model: The model consists of three convolutional and two Maxpolling layers which is then passed to four dense layer.

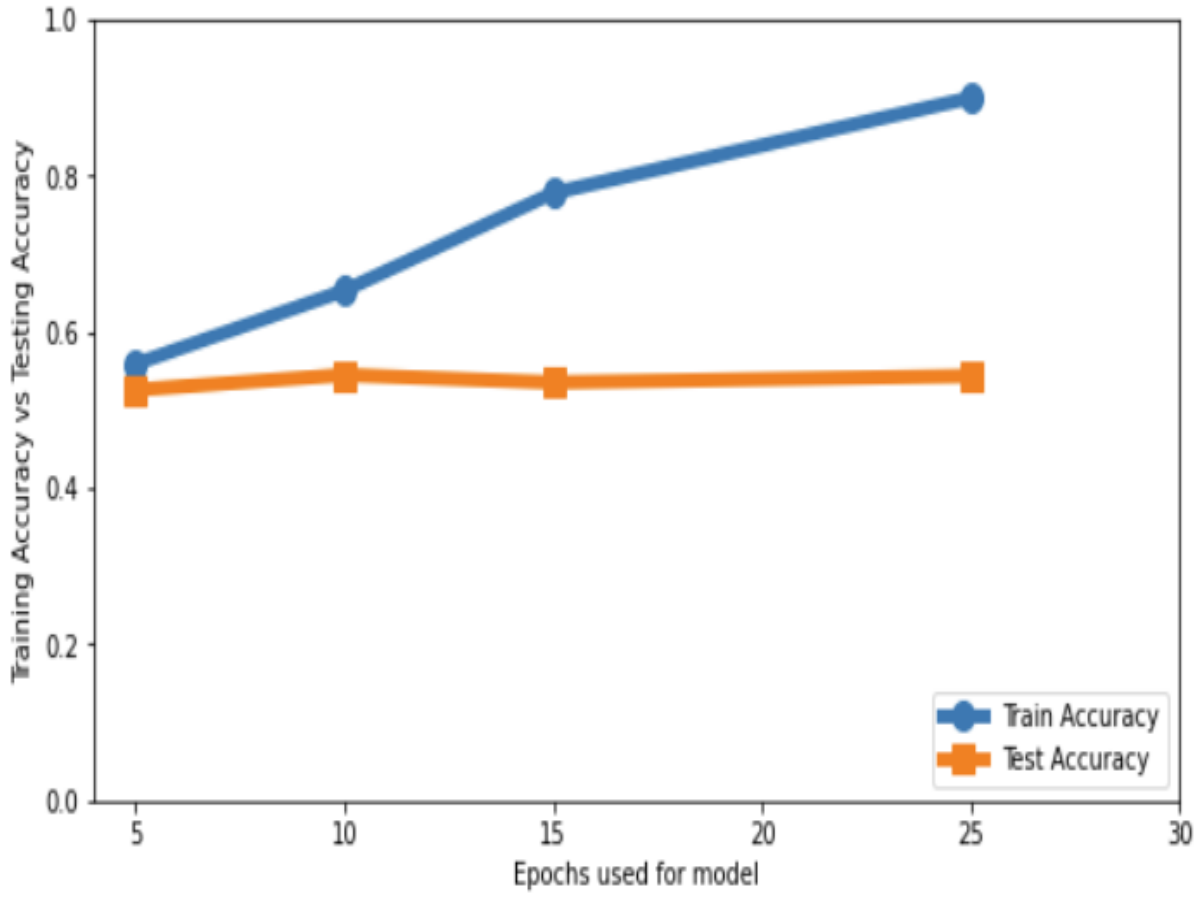
```
[74] model.summary()
```

```
Model: "sequential_10"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_30 (Conv2D)           (None, 46, 46, 32)        320
_________________________________________________________________
max_pooling2d_20 (MaxPooling (None, 23, 23, 32)        0
_________________________________________________________________
conv2d_31 (Conv2D)           (None, 21, 21, 64)        18496
_________________________________________________________________
max_pooling2d_21 (MaxPooling (None, 10, 10, 64)        0
_________________________________________________________________
conv2d_32 (Conv2D)           (None, 8, 8, 64)          36928
_________________________________________________________________
flatten_10 (Flatten)         (None, 4096)              0
_________________________________________________________________
dense_39 (Dense)             (None, 32)                131104
_________________________________________________________________
dense_40 (Dense)             (None, 32)                1056
_________________________________________________________________
dense_41 (Dense)             (None, 32)                1056
_________________________________________________________________
dense_42 (Dense)             (None, 7)                 231
=================================================================
Total params: 189,191
Trainable params: 189,191
Non-trainable params: 0
```

The training data is fitted into the model with an epoch of 10 and the training accuracy obtained is 63% while the loss is 9.6%. This model is tested with the test data, which yields the accuracy of 53% and loss of 12%.

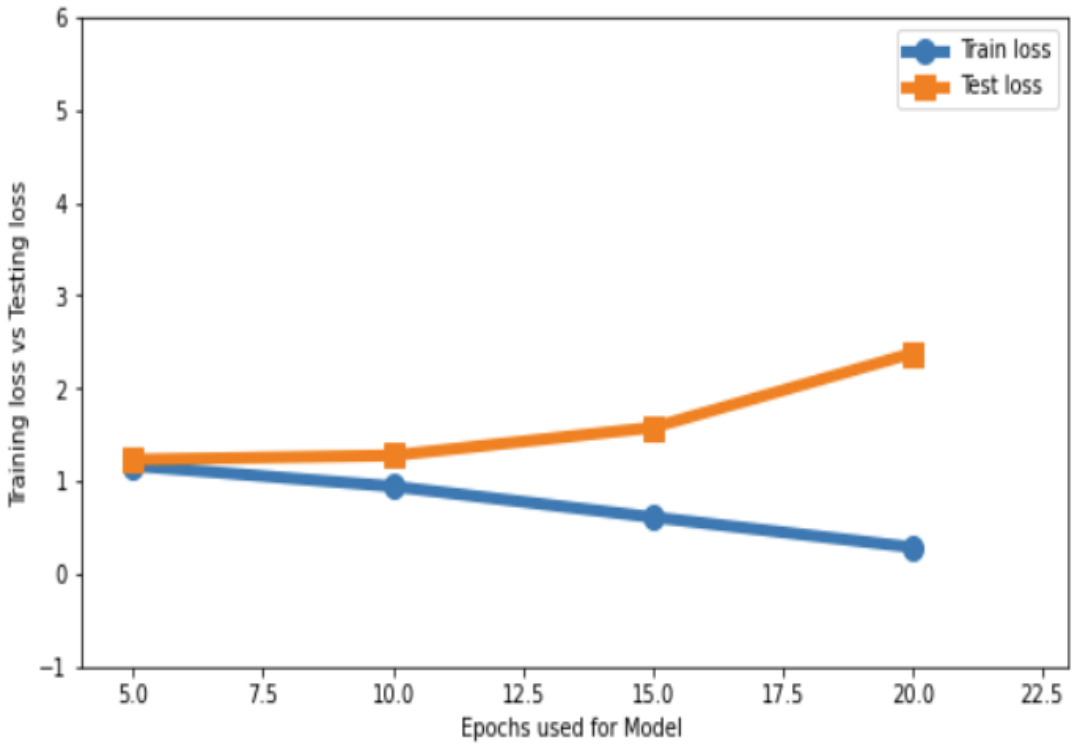
The plot for training accuracy and test accuracy is show below:

```
plt.figure(figsize=(9,5))
plt.plot([5,10,15,25], train_acc, linewidth=5, marker='o', markersize=10)
plt.plot([5,10,15,25], test_acc, linewidth=5, marker='s', markersize=10)
plt.xlabel('Epochs used for model', fontsize=10)
plt.ylabel('Training Accuracy vs Testing Accuracy', fontsize=10)
plt.legend(['Train Accuracy','Test Accuracy'], fontsize=10, loc='lower right')
plt.axis([4, 30, 0, 1])
plt.show()
```



The plot for training loss and testing loss is show below:

```
plt.figure(figsize=(9,5))
plt.plot([5,10,15,20], train_loss, linewidth=5, marker='o', markersize=10)
plt.plot([5,10,15,20], test_loss, linewidth=5, marker='s', markersize=10)
plt.xlabel('Epochs used for Model ', fontsize=10)
plt.ylabel('Training loss vs Testing loss', fontsize=10)
plt.legend(['Train loss','Test loss'], fontsize=10, loc='upper right')
plt.axis([4, 23, -1, 6])
plt.show()
```



Iteration	Parameters	Training and Test Accuracy
1	No. of layers: 3 Conv 2D layers,2 Maxpolling layers and 4 dense layers Number of kernels: 32, 64, 64 No. of neuron in last dense layer: 7 Activation function: ReLu, Last Layer: Softmax Error function: Categorical cross entropy Batch Size: 64 No. of epochs: 5	Train: 53% Test: 52%

2	No. of layers: 3 Conv 2D layers,2 Maxpolling layers and 4 dense layers Number of kernels: 32,64,64 No. of neuron in last dense layer: 7 Activation function: Relu, Last Layer: softmax Error function: Categorical cross entropy Batch Size: 64 No. of epochs: 10	Train: 54.5% Test: 54.4%
3	No. of layers: 3 Conv 2D layers,2 Maxpolling layers and 4 dense layers Number of kernels: 32,64,64 No. of neuron in last dense layer: 7 Activation function: Relu, Last Layer: softmax Error function: Categorical cross entropy Batch Size: 64 No. of epochs: 15	Train: 70% Test: 53.3%
4	No. of layers: 3 Conv 2D layers,2 Maxpolling layers and 4 dense layers Number of kernels: 32,64,64 No. of neuron in last dense layer: 7 Activation function: Relu, Last Layer: softmax Error function: Categorical cross entropy Batch Size: 64 No. of epochs: 20	Train: Test: 54%

Out of the three methods, Convolutional layers performed the based. The accuracy of this model is much higher than the other two. Since the accuracy for this model is still less than optimal accuracy, we can perform fine tuning and image augmentation to increase the accuracy.