LOKESWARI UMAKANTHAN LXU190000
ASHIKA PRAKASH ACHARYA AXA190084
SRIPRASATH GUJULUVA SXG180154

CS6301

Project 1

Deep Learning

Note: Used two free days.

Dataset : Challenges in Representation Learning: Facial Expression Recognition Challenge

Data Source : Facial Expression Recognition Dataset from Kaggle. icml_face_data.csv is used for the training and testing data extraction.

1) Emotion Category : Emotion Category ranges from 0 to 6 . The task is to categorize emotions into one of these seven categories.

- 0=Angry
- 1=Disgust
- 2=Fear
- 3=Happy
- 4=Sad
- 5=Surprise
- 6=Neutral

2) The second column consists of whether the data is a Training data or Test data.

3) The third column consists of the pixels data which are the flattened pixels values of each image that are to be passed as training data or teat data. There are 28709 training data and 7178 Test data (public_test and private_test) of flattened pixel values available for training and testing the model.

Sample training data image

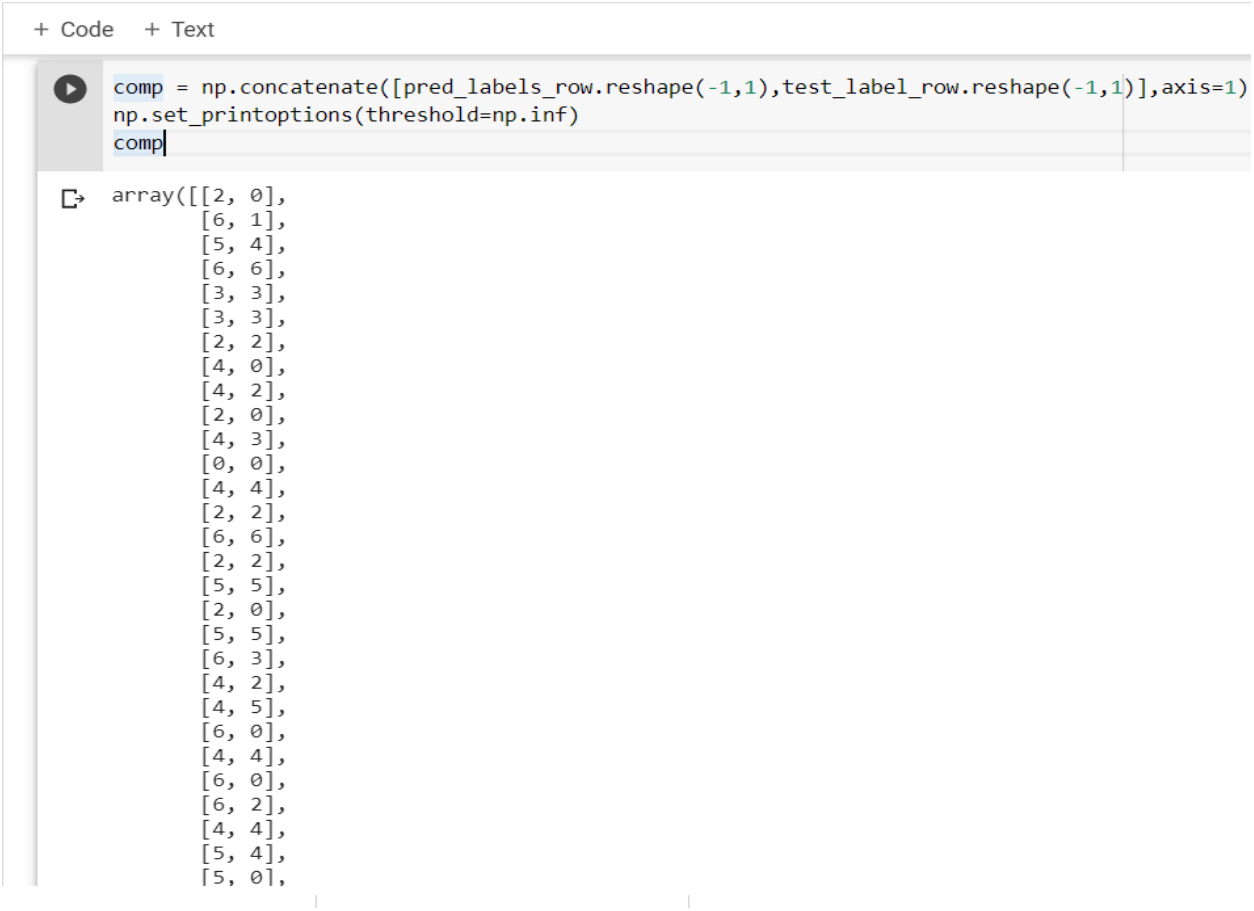
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	emotion	Usage	pixels												
2	0	Training	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 119 115 110 98 91 84 84 90 99 110 126 143 153 158 171 169 172 169 165 129 110 113 107 9
3	0	Training	151 150 147 155 148 133 111 140 170 174 182 154 153 164 173 178 185 185 189 187 186 193 194 185 183 186 180 173 166 161 147 133 17
4	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 21
5	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 1
6	6	Training	4 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
7	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 15
8	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 1
9	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
10	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 13
11	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
12	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 1
13	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
14	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
15	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

Sample Test Data Image

28711	0	PublicTest	254 254 254 254 254 249 255 160 2 58 53 70 77 76 75 78 68 18 32 29 0 54 73 75 72 68 75 77 76 76 75 80 51 36 47 40 44 42 37 48 40 64 54 5
28712	1	PublicTest	156 184 198 202 204 207 210 212 213 214 215 214 214 213 216 217 218 217 216 214 213 214 213 214 215 211 207 205 204 202 198 195 19
28713	4	PublicTest	69 118 61 60 96 121 103 87 103 88 70 90 115 122 123 124 129 132 133 131 131 121 113 110 101 100 99 114 113 105 106 107 120 123 124 1
28714	6	PublicTest	205 203 236 157 83 158 120 116 94 86 155 180 205 231 219 217 190 198 208 174 159 167 211 230 215 209 195 210 202 186 187 182 185 22
28715	3	PublicTest	87 79 74 66 74 96 77 80 80 84 83 89 102 91 84 102 108 107 102 89 96 128 152 176 195 207 214 220 222 224 222 220 216 214 205 197 179 1
28716	3	PublicTest	235 233 223 109 34 37 34 31 28 38 56 69 106 136 153 163 145 135 136 127 152 158 152 144 138 121 65 38 56 42 34 31 28 33 35 39 40 34 29
28717	2	PublicTest	71 70 104 147 166 170 195 145 156 154 146 129 139 130 117 103 104 107 111 101 90 79 75 110 126 101 79 89 95 113 107 111 105 102 129
28718	0	PublicTest	176 177 170 168 173 171 167 169 166 139 98 107 121 136 138 141 154 155 160 155 153 143 135 121 101 103 101 76 27 13 17 17 17 22 45 5
28719	2	PublicTest	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
28720	0	PublicTest	126 126 123 119 116 113 112 111 110 111 93 72 107 109 127 166 190 203 206 209 209 210 211 210 203 199 194 183 173 160 142 121 83 71
28721	3	PublicTest	180 175 169 161 157 158 157 154 155 157 162 169 168 165 159 153 150 149 150 151 153 155 163 166 169 170 170 176 179 180 183 188 18
28722	0	PublicTest	88 46 35 27 22 32 59 59 62 76 136 148 153 126 109 108 92 90 103 118 117 121 130 115 90 88 87 87 85 103 128 110 88 72 54 54 50 34 29 6 1
28723	4	PublicTest	121 112 64 104 101 87 118 74 91 128 89 109 91 27 127 197 191 186 189 192 194 197 195 192 190 186 178 177 169 161 151 133 105 84 52 2
28724	2	PublicTest	165 203 211 204 216 204 202 194 195 191 207 209 196 202 209 214 214 215 193 186 175 128 180 208 160 130 189 215 188 169 184 201 18
28725	6	PublicTest	22 28 27 28 26 28 31 33 33 30 32 23 19 44 75 110 120 127 138 138 151 155 129 128 127 125 118 117 119 96 109 94 65 54 18 13 22 26 26 27
28726	2	PublicTest	132 154 165 176 182 187 192 199 203 206 208 212 216 221 223 223 221 222 223 222 220 218 217 214 209 207 210 212 206 202 197 192 19
28727	5	PublicTest	255 255 255 254 254 252 174 59 35 34 40 57 84 111 129 139 147 155 161 165 171 174 176 180 183 187 187 186 183 176 169 146 118 84 50
28728	0	PublicTest	159 161 160 157 148 151 137 143 141 140 147 139 126 159 170 154 179 180 183 193 194 167 175 152 170 164 161 164 166 192 190 203 20

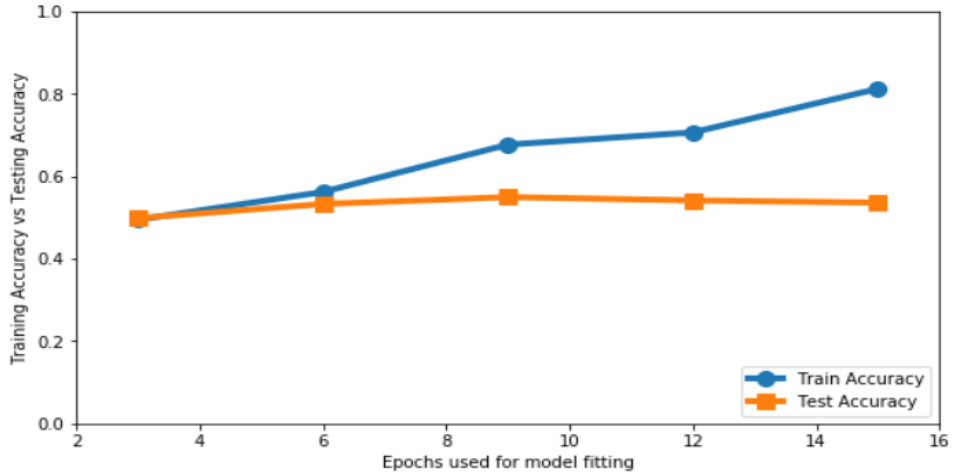
Predicted Labels:

The predicted labels of the 7178 Test data is obtained from the model.predict() function. The output and the true label are compared and the predicted label and true label of 25 samples are displayed.



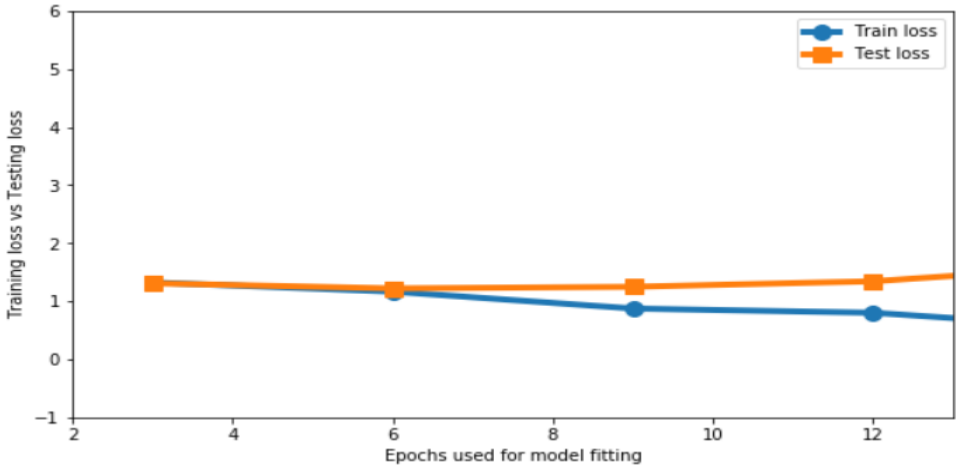
Plotting Training Accuracy and Test Accuracy :

```
plt.figure(figsize=(9,5))
plt.plot([3,6,9,12,15], train_acc, linewidth=4, marker='o', markersize=10)
plt.plot([3,6,9,12,15], test_acc, linewidth=4, marker='s', markersize=10)
plt.xlabel('Epochs used for model fitting', fontsize=10)
plt.ylabel('Training Accuracy vs Testing Accuracy', fontsize=10)
plt.legend(['Train Accuracy','Test Accuracy'], fontsize=10, loc='lower right')
plt.axis([2, 16, 0, 1])
plt.show()
```



Plotting Training Loss and Test Loss :

```
plt.figure(figsize=(9,5))
plt.plot([3,6,9,12,15], train_loss, linewidth=4, marker='o', markersize=10)
plt.plot([3,6,9,12,15], test_loss, linewidth=4, marker='s', markersize=10)
plt.xlabel('Epochs used for model fitting', fontsize=10)
plt.ylabel('Training loss vs Testing loss', fontsize=10)
plt.legend(['Train loss','Test loss'], fontsize=10, loc='upper right')
plt.axis([2, 13, -1, 6])
plt.show()
```



Parameter Testing and Tuning:

ITERATION	PARAMETERS	TRAINING AND TEST ACCURACY
1.	1. No of layers = • 3 conv 2D layers • 2 max pooling layers • After flattening, 4 dense layers 2. Kernel size = 3 * 3 3. No of kernels in each layer = • Conv 2D layer 1 : 32 • Conv 2D layer 2 : 64 • Conv 2D layer 3 : 64 4. No of neurons in the last dense layer = 7 5. Activation function = All layers except last: RELU Last layer : Softmax 6. Error function = Categorical cross entropy 7. Batch size = 64 8. No of epochs =10	Train accuracy : 0.6853 Test accuracy: 0.5626

```
model.fit(train_images, train_labels, epochs=10, batch_size=64)
```

```
Epoch 1/10
28709/28709 [==============================] - 5s 191us/step - loss: 1.7093 - acc: 0.3099
Epoch 2/10
28709/28709 [==============================] - 4s 144us/step - loss: 1.4692 - acc: 0.4298
Epoch 3/10
28709/28709 [==============================] - 4s 144us/step - loss: 1.3301 - acc: 0.4917
Epoch 4/10
28709/28709 [==============================] - 4s 145us/step - loss: 1.2415 - acc: 0.5282
Epoch 5/10
28709/28709 [==============================] - 4s 143us/step - loss: 1.1725 - acc: 0.5554
Epoch 6/10
28709/28709 [==============================] - 4s 145us/step - loss: 1.1075 - acc: 0.5842
Epoch 7/10
28709/28709 [==============================] - 4s 145us/step - loss: 1.0417 - acc: 0.6096
Epoch 8/10
28709/28709 [==============================] - 4s 144us/step - loss: 0.9781 - acc: 0.6337
Epoch 9/10
28709/28709 [==============================] - 4s 144us/step - loss: 0.9123 - acc: 0.6604
Epoch 10/10
28709/28709 [==============================] - 4s 144us/step - loss: 0.8475 - acc: 0.6853
<keras.callbacks.History at 0x7ffae020dfd0>
```

2.	1. No of layers = • 3 conv 2D layers • 3 max pooling layers • After flattening, 4 dense layers 2. Kernel size = 3 * 3 3. No of kernels in each layer = • Conv 2D layer 1 : 32 • Conv 2D layer 2 : 64 • Conv 2D layer 3 : 64 4. No of neurons in the last dense layer = 7 5. Activation function = All layers except last: RELU Last layer : Softmax 6. Error function = Categorical cross entropy 7. Batch size = 64 8. No of epochs =9	Train accuracy : 0.6768 Test accuracy: 0.5495

```
9
Epoch 1/9
28709/28709 [==============================] - 6s 211us/step - loss: 1.6869 - acc: 0.3256
Epoch 2/9
28709/28709 [==============================] - 4s 143us/step - loss: 1.4309 - acc: 0.4479
Epoch 3/9
28709/28709 [==============================] - 4s 142us/step - loss: 1.3047 - acc: 0.4985
Epoch 4/9
28709/28709 [==============================] - 4s 143us/step - loss: 1.2169 - acc: 0.5373
Epoch 5/9
28709/28709 [==============================] - 4s 141us/step - loss: 1.1417 - acc: 0.5664
Epoch 6/9
28709/28709 [==============================] - 4s 142us/step - loss: 1.0716 - acc: 0.5960
Epoch 7/9
28709/28709 [==============================] - 4s 141us/step - loss: 1.0035 - acc: 0.6226
Epoch 8/9
28709/28709 [==============================] - 4s 142us/step - loss: 0.9354 - acc: 0.6506
Epoch 9/9
28709/28709 [==============================] - 4s 141us/step - loss: 0.8748 - acc: 0.6769
7178/7178 [==============================] - 1s 187us/step
0.874770067296119 0.6768957469554616 1.249667315703642 0.5495959877569252
```

3.	1. No of layers = • 3 conv 2D layers • 3 max pooling layers • After flattening, 4 dense layers 2. Kernel size = 3 * 3 3. No of kernels in each layer = • Conv 2D layer 1 : 32 • Conv 2D layer 2 : 64 • Conv 2D layer 3 : 64 4. No of neurons in the last dense layer = 7 5. Activation function = All layers except last: RELU Last layer : Softmax 6. Error function = Categorical cross entropy 7. Batch size = 64 8. No of epochs =15	Train accuracy : 0.8125 Test accuracy: 0.5360

```
Epoch 1/15
28709/28709 [==============================] - 6s 212us/step - loss: 1.7119 - acc: 0.3144
Epoch 2/15
28709/28709 [==============================] - 4s 140us/step - loss: 1.4727 - acc: 0.4280
Epoch 3/15
28709/28709 [==============================] - 4s 140us/step - loss: 1.3329 - acc: 0.4859
Epoch 4/15
28709/28709 [==============================] - 4s 141us/step - loss: 1.2476 - acc: 0.5262
Epoch 5/15
28709/28709 [==============================] - 4s 140us/step - loss: 1.1765 - acc: 0.5566
Epoch 6/15
28709/28709 [==============================] - 4s 139us/step - loss: 1.1063 - acc: 0.5805
Epoch 7/15
28709/28709 [==============================] - 4s 140us/step - loss: 1.0391 - acc: 0.6101
Epoch 8/15
28709/28709 [==============================] - 4s 141us/step - loss: 0.9737 - acc: 0.6376
Epoch 9/15
28709/28709 [==============================] - 4s 140us/step - loss: 0.9052 - acc: 0.6656
Epoch 10/15
28709/28709 [==============================] - 4s 139us/step - loss: 0.8353 - acc: 0.6940
Epoch 11/15
28709/28709 [==============================] - 4s 140us/step - loss: 0.7742 - acc: 0.7178
Epoch 12/15
28709/28709 [==============================] - 4s 141us/step - loss: 0.7074 - acc: 0.7436
Epoch 13/15
28709/28709 [==============================] - 4s 141us/step - loss: 0.6459 - acc: 0.7677
Epoch 14/15
28709/28709 [==============================] - 4s 140us/step - loss: 0.5817 - acc: 0.7922
Epoch 15/15
28709/28709 [==============================] - 4s 140us/step - loss: 0.5256 - acc: 0.8125
7178/7178 [==============================] - 1s 191us/step
0.5256078512679655 0.8125326552565948 1.623607430427511 0.536082474230956
```

CONCLUSION:

- Achieved better accuracy with increase in epochs from 3 to 15. Increasing more number of epochs than 15 improves the training accuracy but do not have any effect on test accuracy. This leads to Overfitting of the model.
- The Number of Conv2D, max-pooling and Dense layers affects the loss and accuracy. The loss and accuracy is measured by Categorical_Crossentropy and the Optimizer is rmsprop (Root Mean Square Propagation).
- Model has been trained by varying epochs from 3 to 15 and calculated their respective training, test loss and accuracy. This information has been included in the Parameter testing and tuning.
- An array of predicted labels and its respective true labels have been concatenated together for comparison. The Corresponding plots of training loss & accuracy and testing loss & accuracy information has been plotted using matplotlib.