Facial Emotion Recognition

By:
Radhika Kulkarni BT15CSE064
Shreyas Dikshit BT15CSE082

Problem Statement:

Detect different types of emotions displayed by humans

What is emotion detection?

- Process of identifying human emotion from facial expressions as well as from verbal expressions. This is both something that humans do automatically but computational methodologies have also been developed.
- The task of emotion recognition often involves the analysis of human expressions in multimodal forms such as texts, audio, or video.

- Different emotion types are detected through the integration of information from facial expressions, body movement and gestures, and speech.

- The technology is said to contribute in the emergence of the so-called emotional or emotive Internet.

Importance of Emotion Detection

- 90% of communication can be non verbal according to researchers.
- However, technology is struggling to keep up. It is difficult to analyse the various intonations and intentions.
- Many countries are speculating the adaption of emotion recognizers in key areas of a city like airports, railway stations, markets, etc.
- This is to automatically analyse emotions and the threats displayed
- This will result in reduction of anti-social activities and maintain harmony.

Importance of Emotion Detection

- Global IT companies like Google and Facebook have invested in startups dedicated to detection of emotion and human physiological signals.
- The emotion recognition solution market is worth USD 6.7 billion in 2016 and estimated to reach USD 67.1 billion by 2021

How to detect emotions?

There are various approaches used to detect emotions based on the available data, namely:

- Knowledge-based techniques- utilize domain knowledge and the semantic and syntactic characteristics of language in order to detect certain emotion types.
- Statistical-based techniques- commonly involve Machine Learning techniques eg. SVM, Naive Bayes; Deep-learning algos like CNN etc.

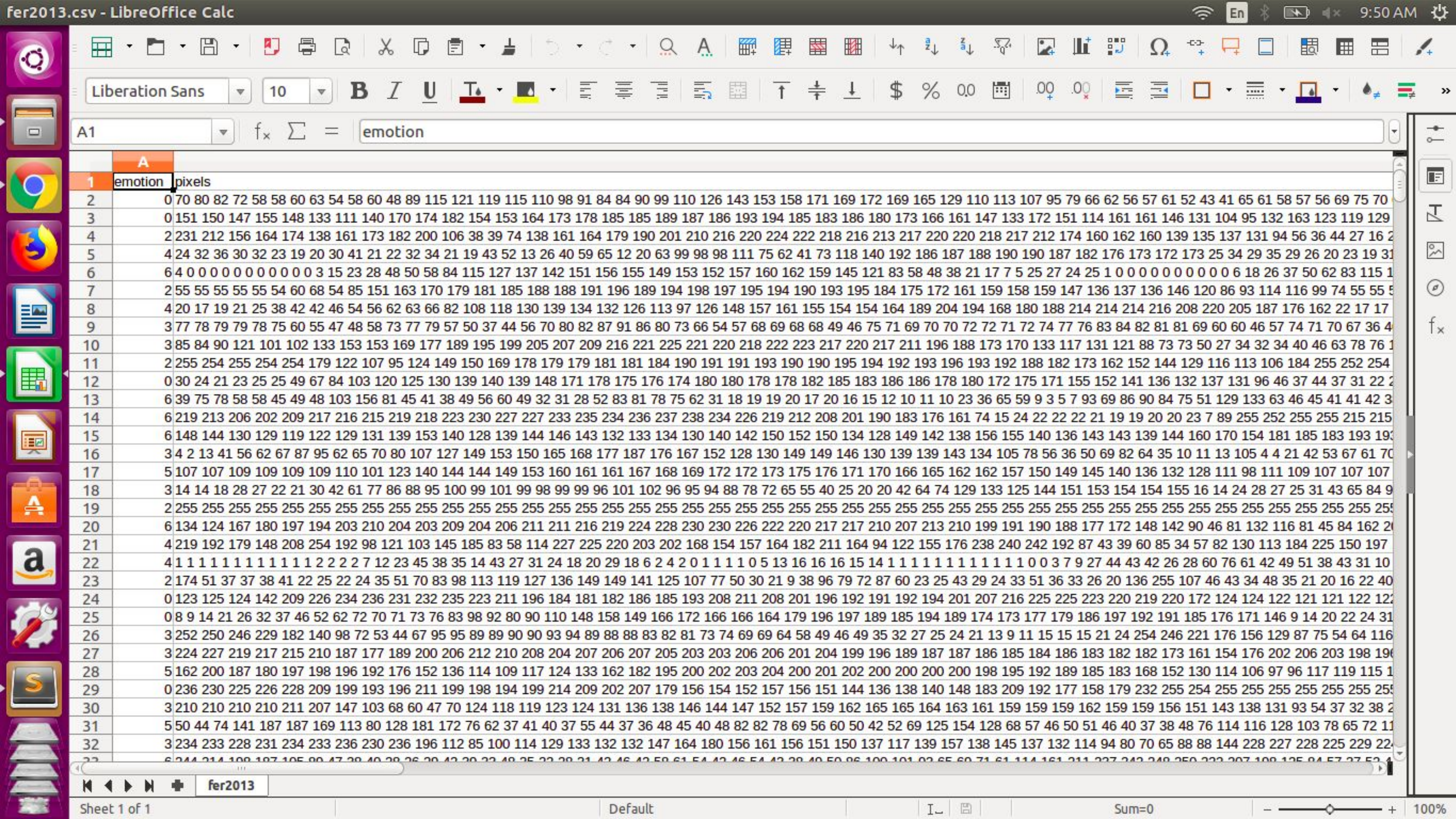
Technology Stack

Programming Language used: Python

- NumPy
- Pandas
- OpenCV
- Keras
- Theano

Data Set

- The data used is an open source dataset from a Kaggle competition
- It involves image pixel values stored in a csv file
- Images are 48 x 48 pixels
- It has labels for 7 different emotions mainly displayed by humans
- These are **0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral**
- The data set consists of 28,709 examples

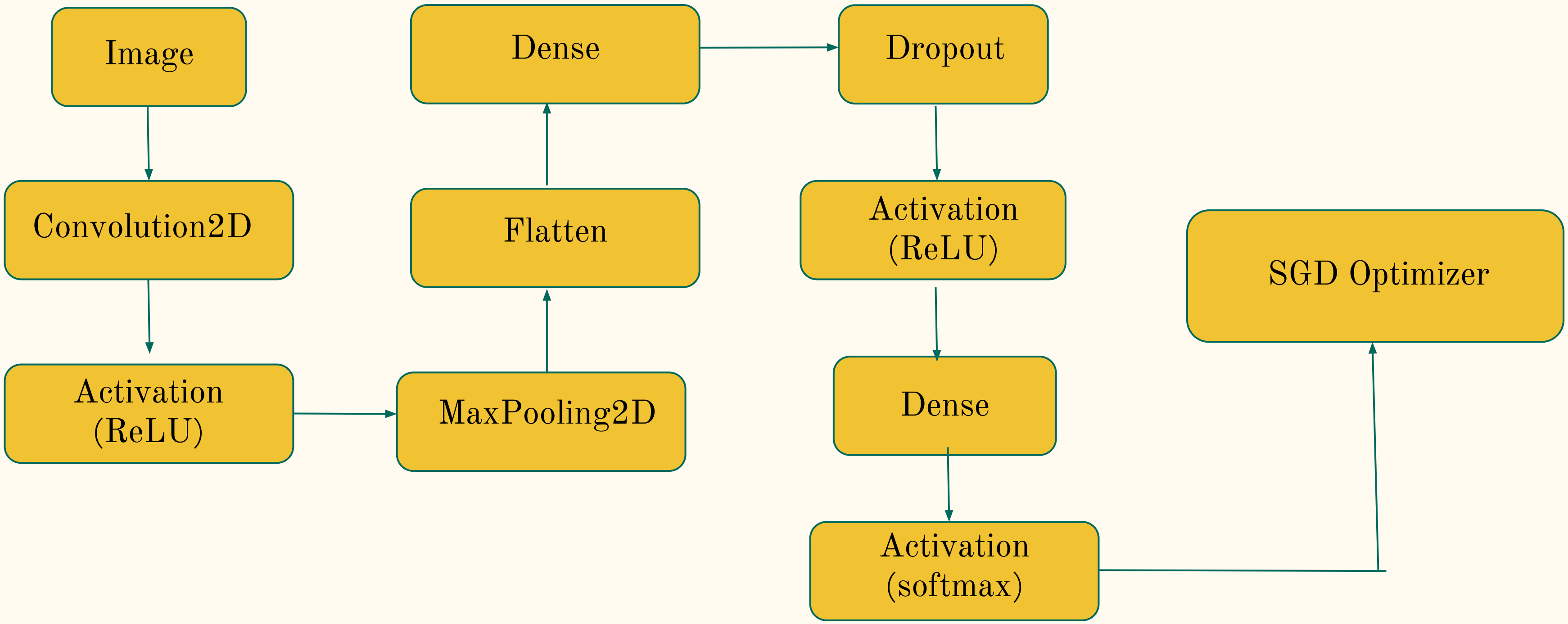


	A	
1	emotion	pixels
2	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 119 115 110 98 91 84 84 90 99 110 126 143 153 158 171 169 172 169 165 129 110 113 107 95 79 66 62 56 57 61 52 43 41 65 61 58 57 56 69 75 70
3	0	151 150 147 155 148 133 111 140 170 174 182 154 153 164 173 178 185 185 189 187 186 193 194 185 183 186 180 173 166 161 147 133 172 151 114 161 161 146 131 104 95 132 163 123 119 129
4	2	231 212 156 164 174 138 161 173 182 200 106 38 39 74 138 161 164 179 190 201 210 216 220 224 222 218 216 213 217 220 220 218 217 212 174 160 162 160 139 135 137 131 94 56 36 44 27 16 2
5	4	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 73 118 140 192 186 187 188 190 190 187 182 176 173 172 173 25 34 29 35 29 26 20 23 19 31
6	6	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 24 25 1 0 0 0 0 0 0 0 0 0 6 18 26 37 50 62 83 115 1
7	2	55 55 55 55 55 54 60 68 54 85 151 163 170 179 181 185 188 188 191 196 189 194 198 197 195 194 190 193 195 184 175 172 161 159 158 159 147 136 137 136 146 120 86 93 114 116 99 74 55 55 5
8	4	20 17 19 21 25 38 42 42 46 54 56 62 63 66 82 108 118 130 139 134 132 126 113 97 126 148 157 161 155 154 154 164 189 204 194 168 180 188 214 214 214 216 208 220 205 187 176 162 22 17 17
9	3	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 84 82 81 81 69 60 60 46 57 74 71 70 67 36 4
10	3	85 84 90 121 101 102 133 153 153 169 177 189 195 199 205 207 209 216 221 225 221 220 218 222 223 217 220 217 211 196 188 173 170 133 117 131 121 88 73 73 50 27 34 32 34 40 46 63 78 76 1
11	2	255 254 255 254 254 179 122 107 95 124 149 150 169 178 179 179 181 181 184 190 191 191 193 190 190 195 194 192 193 196 193 192 188 182 173 162 152 144 129 116 113 106 184 255 252 254
12	0	30 24 21 23 25 25 49 67 84 103 120 125 130 139 140 139 148 171 178 175 176 174 180 180 178 178 182 185 183 186 186 178 180 172 175 171 155 152 141 136 132 137 131 96 46 37 44 37 31 22 2
13	6	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 69 86 90 84 75 51 129 133 63 46 45 41 41 42 3
14	6	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 21 19 19 20 20 23 7 89 255 252 255 255 215 215
15	6	148 144 130 129 119 122 129 131 139 153 140 128 139 144 146 143 132 133 134 130 140 142 150 152 150 134 128 149 142 138 156 155 140 136 143 143 139 144 160 170 154 181 185 183 193 19
16	3	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 69 82 64 35 10 11 13 105 4 4 21 42 53 67 61 70
17	5	107 107 109 109 109 109 110 101 123 140 144 144 149 153 160 161 161 167 168 169 172 172 173 175 176 171 170 166 165 162 162 157 150 149 145 140 136 132 128 111 98 111 109 107 107 107
18	3	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 74 129 133 125 144 151 153 154 154 155 16 14 24 28 27 25 31 43 65 84 9
19	2	255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 255 25
20	6	134 124 167 180 197 194 203 210 204 203 209 204 206 211 211 216 219 224 228 230 230 226 222 220 217 217 210 207 213 210 199 191 190 188 177 172 148 142 90 46 81 132 116 81 45 84 162 20
21	4	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 43 39 60 85 34 57 82 130 113 184 225 150 197
22	4	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 0 0 3 7 9 27 44 43 42 26 28 60 76 61 42 49 51 38 43 31 10
23	2	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 20 136 255 107 46 43 34 48 35 21 20 16 22 40
24	0	123 125 124 142 209 226 234 236 231 232 235 223 211 196 184 181 182 186 185 193 208 211 208 201 196 192 191 192 194 201 207 216 225 225 223 220 219 220 172 124 124 122 121 121 122 12
25	0	8 9 14 21 26 32 37 46 52 62 72 70 71 73 76 83 98 92 80 90 110 148 158 149 166 172 166 166 164 179 196 197 189 185 194 189 174 173 177 179 186 197 192 191 185 176 171 146 9 14 20 22 24 31
26	3	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 24 254 246 221 176 156 129 87 75 54 64 116
27	3	224 227 219 217 215 210 187 177 189 200 206 212 210 208 204 207 206 207 205 203 203 206 206 201 204 199 196 189 187 187 186 185 184 186 183 182 182 173 161 154 176 202 206 203 198 19
28	5	162 200 187 180 197 198 196 192 176 152 136 114 109 117 124 133 162 182 195 200 202 203 204 200 201 202 200 200 200 200 198 195 192 189 185 183 168 152 130 114 106 97 96 117 119 115 1
29	0	236 230 225 226 228 209 199 193 196 211 199 198 194 199 214 209 202 207 179 156 154 152 157 156 151 144 136 138 140 148 183 209 192 177 158 179 232 255 254 255 255 255 255 255 255 25
30	3	210 210 210 210 211 207 147 103 68 60 47 70 124 118 119 123 124 131 136 138 146 144 147 152 157 159 162 165 165 164 163 161 159 159 159 162 159 159 156 151 143 138 131 93 54 37 32 38 2
31	5	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 46 40 37 38 48 76 114 116 128 103 78 65 72 11
32	3	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 94 80 70 65 88 88 144 228 227 228 225 229 22

Preprocessing

- The only preprocessing operation performed in “normalization”.
- All discreet pixel values are normalized to [0 , 255].

CNN architecture



Image

Convolution2D

Activation (ReLU)

MaxPooling2D

Flatten

Dense

Dropout

Activation (ReLU)

Dense

Activation (softmax)

SGD Optimizer

CNN architecture

- **Convolution2D**: Convolution operator for filtering windows of two-dimensional inputs. No. of convolution filters: 32, No. of rows/columns in convolution kernel: 3
- **Activation (ReLU)** : ReLU is the most used activation function in convolutional neural networks. f(z) = 0 when z < 0 and f(z) = z when z>= 0.
- **MaxPooling2D** : We have a 2x2 filter that we'll run over our input. For each of the regions, we will take **max** of that region and create an output matrix where each element is the max of a region in the original input.
- **Flatten** : Flattens the input. Does not affect the batch size. From m×n to (mn)×1
- **Dense** : dense layer with 128 outputs.

CNN architecture (contd.)

- **Dropout**: randomly drops ‘p’ fraction (in this case 0.4) of i/p units at each update during training time, which helps prevent overfitting.
- **Activation(softmax):** function will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs. The main advantage of using Softmax is the output probabilities range. The range will be **0 to 1.**
- **Stochastic Gradient Descent Optimizer**

Haar Features

- Haar-like features are used in any kind of object detection.
- These were used by Viola and Jones for the first real time face detector.
- A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums.
- This difference is then used to categorize subsections of an image
- In detection phase, a window is slided across the image. For each subsection, haar-like features are calculated.
- Difference is compared to a learned threshold that separated objects from non objects.

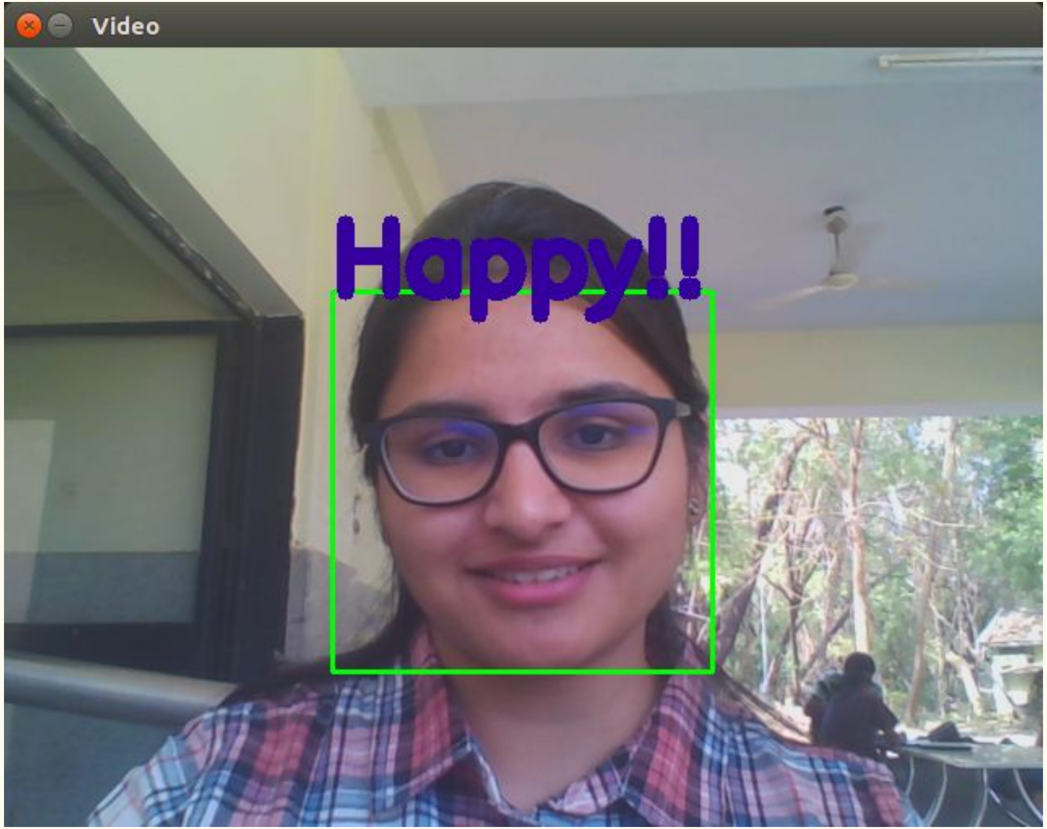
Face Detection using Haar Cascade

- We first need to detect a face. For this, we are using haar cascade.
- The haarcascade_frontalface_default.xml is a haar cascade designed by OpenCV to detect the frontal face.
- It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images.
- OpenCV comes with its own trainer and detector.
- We load the classifier and give an input which is in grayscale form. The location of the detected faces is returned in the form of a rectangle with coordinates x, y, w, h.

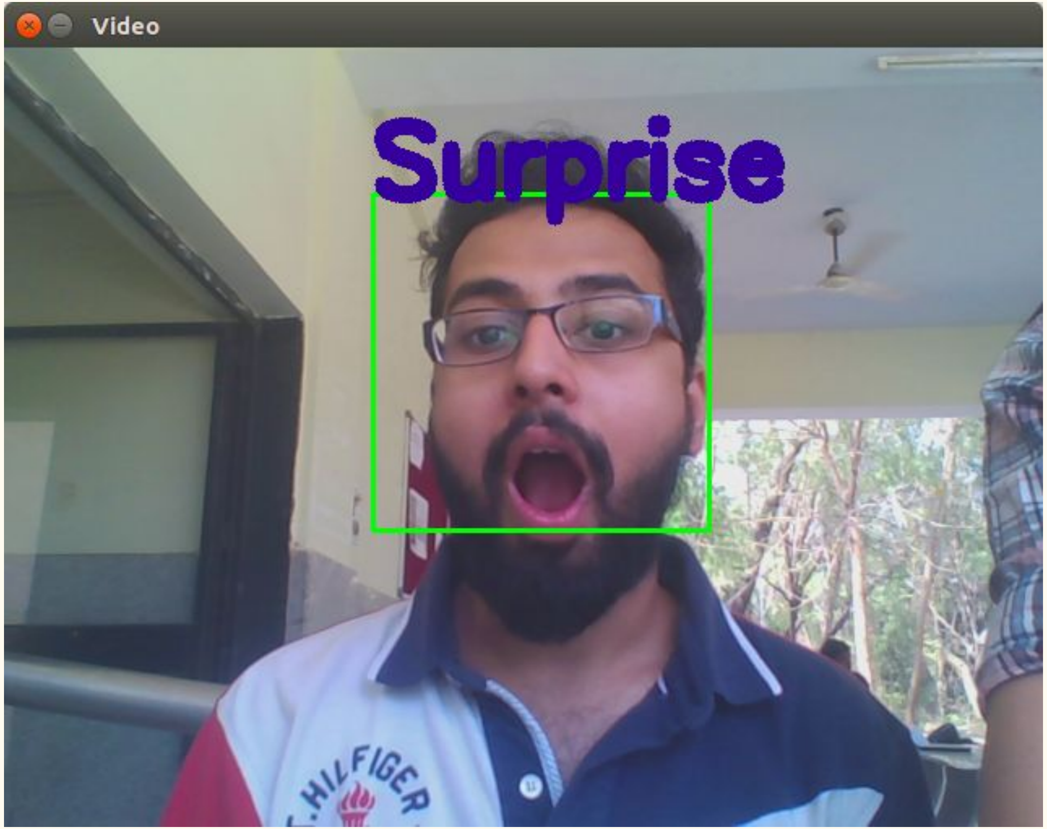
Working model

The following are the steps:

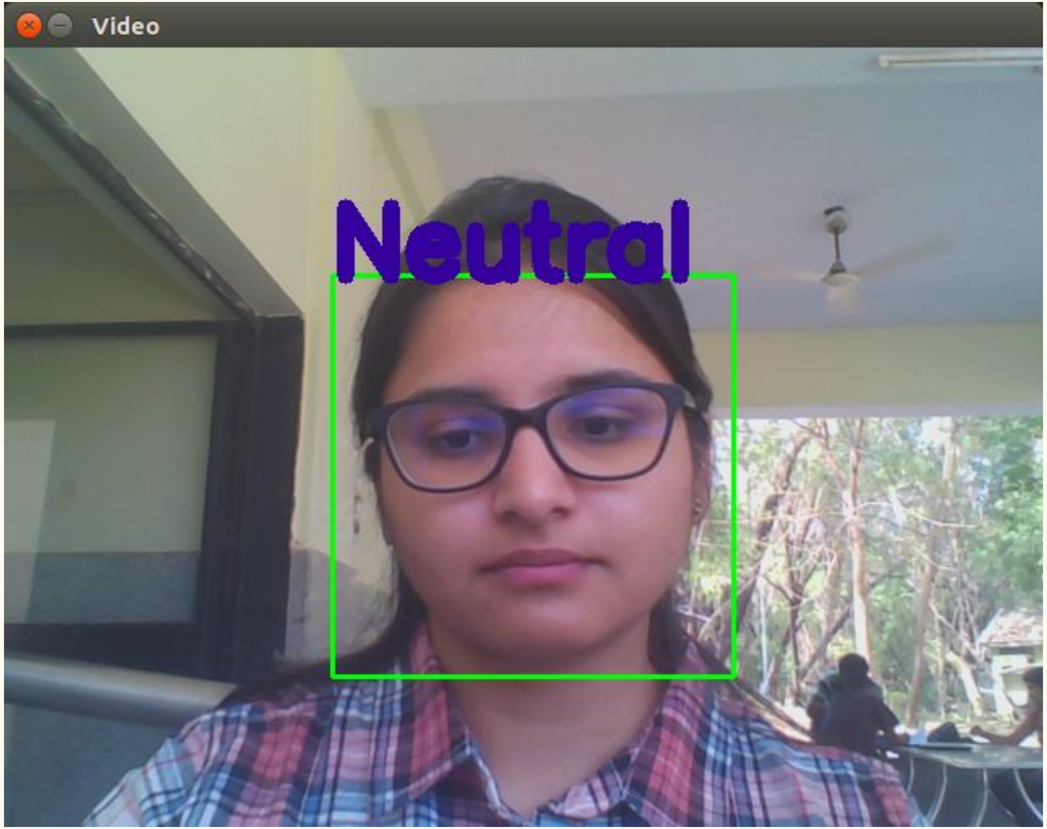
- Load the fer2013 dataset and normalize it.
- The neural network is built and trained with batch size 150 and number of epochs as 15.
- The model architecture is stored and the weights are also stored.
- OpenCV’s VideoCapture function is used to first capture video frames from web camera.
- These frames are first used to detect faces.
- Detected faces are then given to the model to predict the output.



Classified as happy



Classified as Surprise



Classified as neutral

References

- Real time CNN for emotion and gender classification by Arriaga, Ploger, Valdenegro
- https://docs.opencv.org/3.4.3/d7/d8b/tutorial_py_face_detection.html
- http://faroit.com/keras-docs/1.2.2/
- https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data
- https://en.wikipedia.org/wiki/Emotion_recognition
-