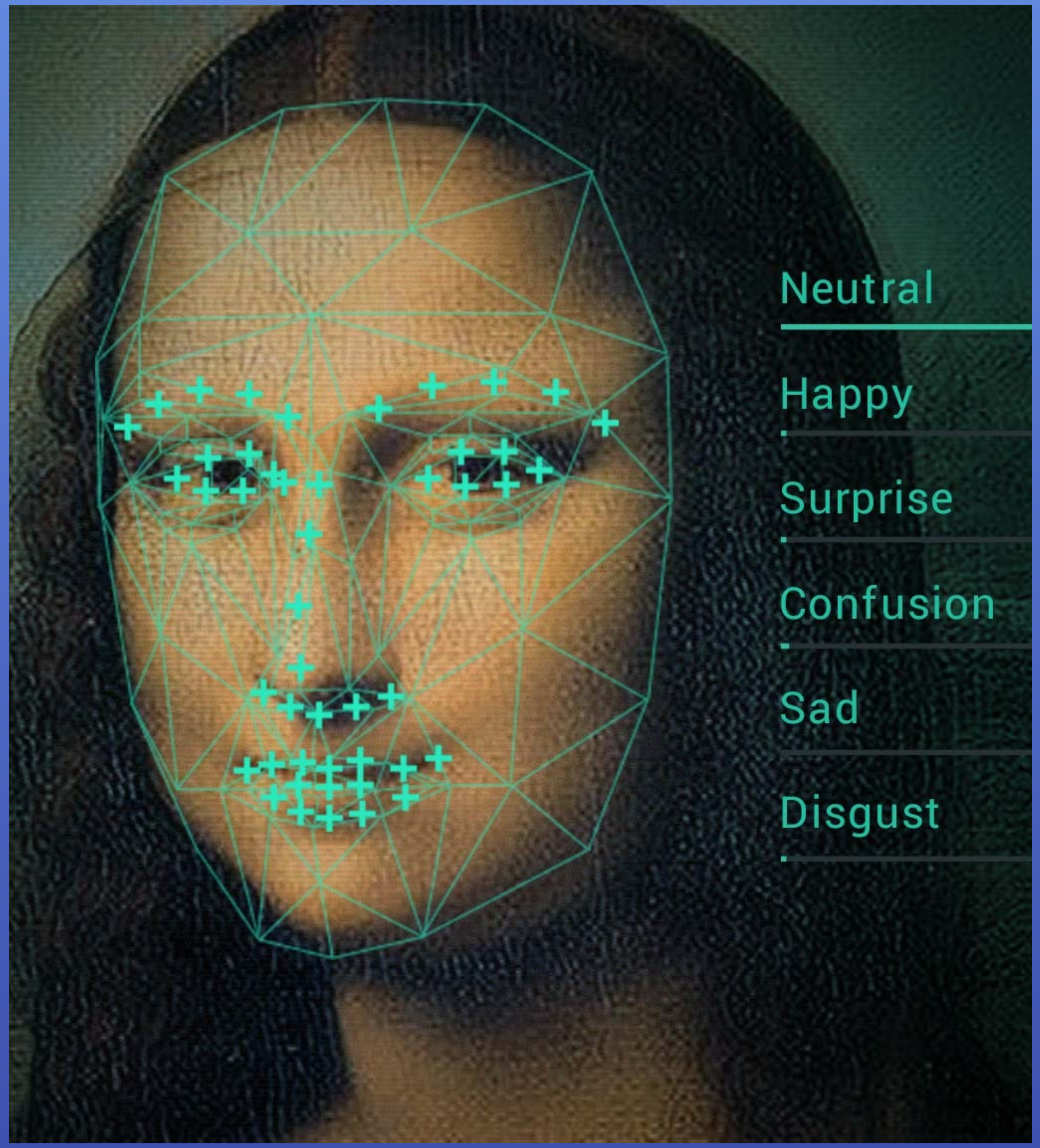


Facial Emotional Recognization

GRETA VATKAR

Introduction

A human face has significant and distinguishing characteristics that aid in the recognition of facial expressions. FER is defined as a change in facial expression caused by an individual's internal emotional state. It is used in a wide range of human-computer interaction (HCI) applications.

The ability to detect and extract human emotions is critical for efficient human machine interaction and provide better service.FER is used in wide range of computer human interaction applications like

Applications

Health Care

FER is used to monitor mental health and therapy to understand patients' emotional states. Based on patients emotional response therapist can analyze behavior pattern and provide more efficient customized therapy

Customer Service

Understanding customer behavior enabled organizations, to deliver empathy, reduce the pressure, and provide more memorable experiences

Hospitality

Hospitality operators and restauranteurs, can greatly benefit by offering personalized customer experiences.

Problem Statement / Models

Problem Statement

Detect human behavior / mood based on facial expression using image data points.

Models

Support Vector Machines (SVMs):SVMs are used for classification tasks, and they can be applied to facial emotion detection by mapping facial features to a decision boundary that distinguishes between "happy" and "not happy" expressions.

SVMs might not capture complex non-linear relationships in the data as effectively as some deep learning models.

Convolutional Neural Networks (CNNs):

This is popular model for image processing

- CNNs are effective at capturing spatial features in facial expressions, making them suitable for image-based tasks like emotion detection.
- They can automatically learn relevant features from the input data, reducing the need for manual feature engineering.
- Transfer learning with pre-trained CNNs on large datasets can be beneficial for facial emotion detection.

Understanding Data

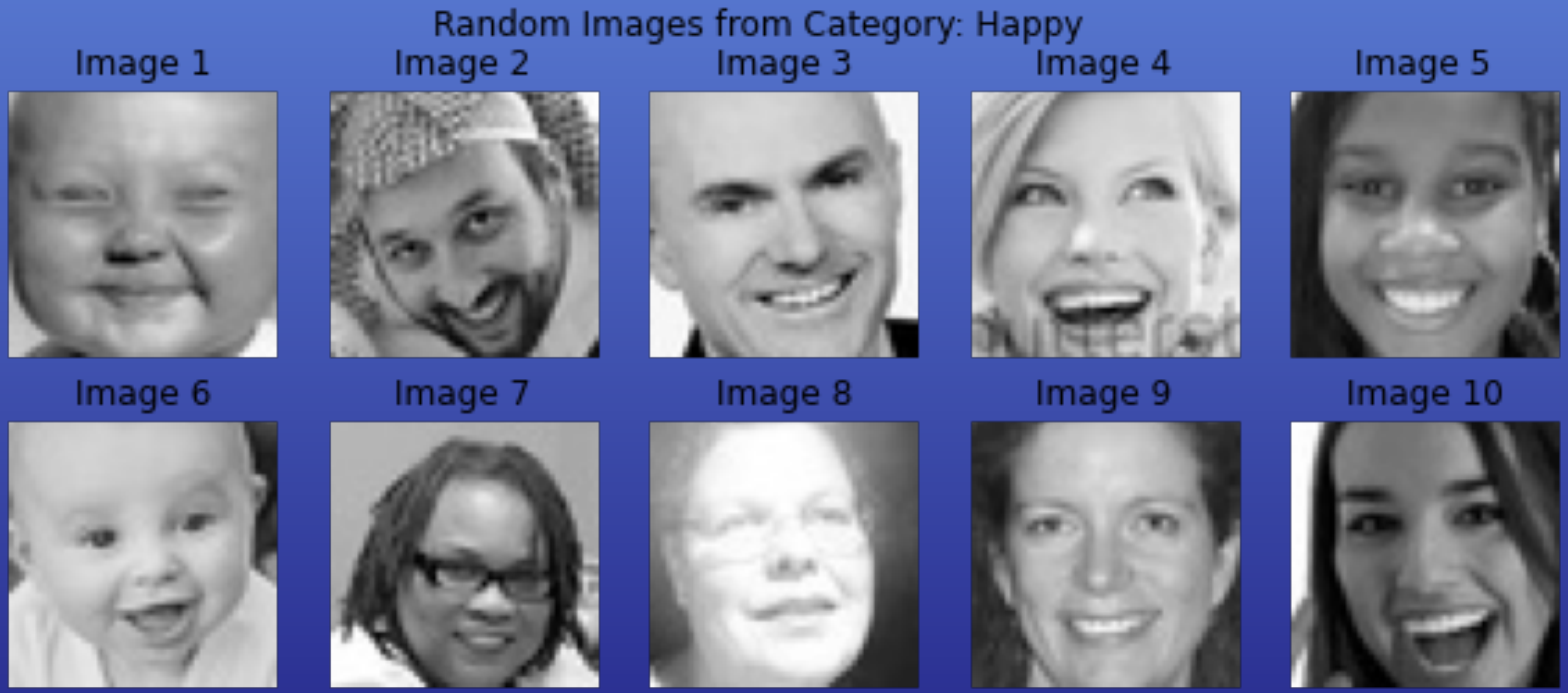
The name of the data set is fer2013 which is an open-source data set that was made publicly available for a Kaggle. It contains 4446 48x48 pixel grayscale facial images and it is not full fledged data.

Data Collection

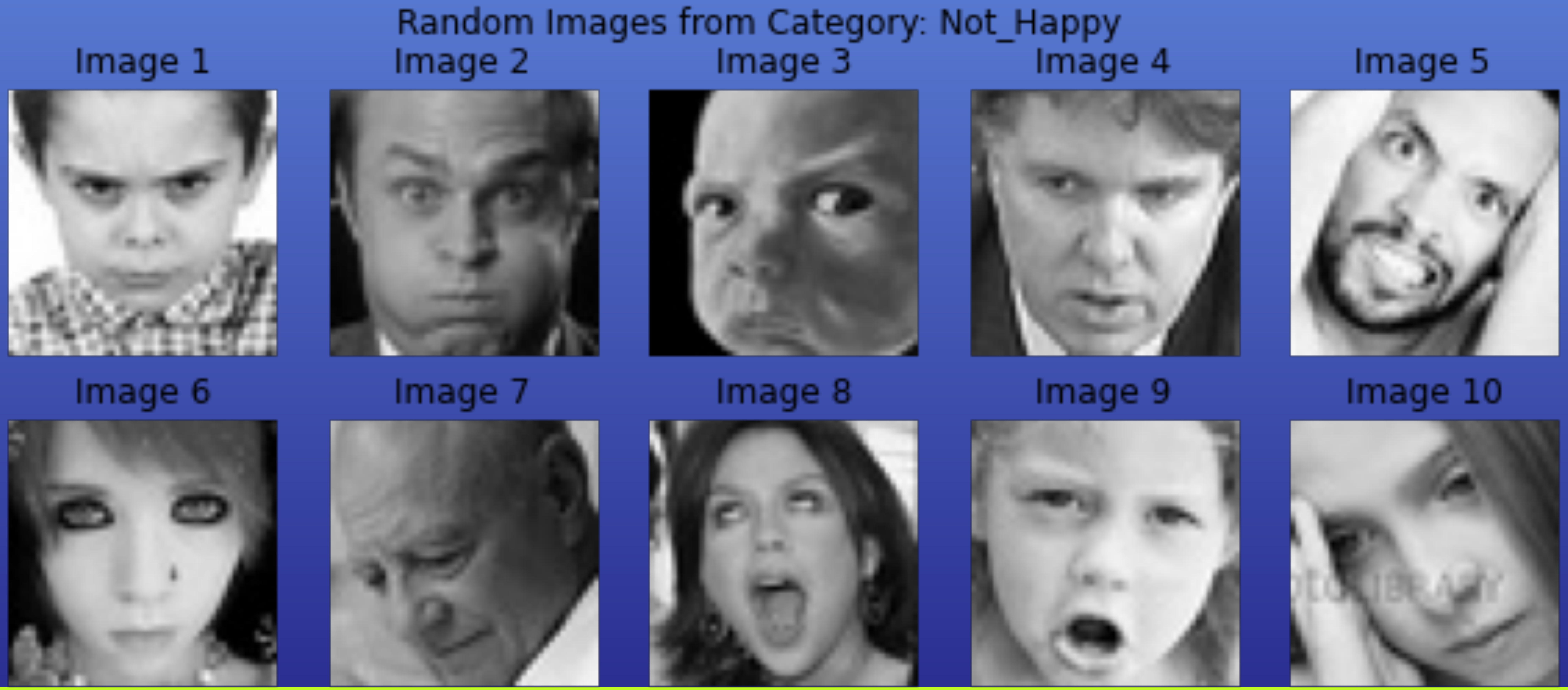
The images are sourced from variety of online platforms like Google search
Measures were implemented to make sure accuracy of label. Some data due to ambiguity or noise might have inaccurate label

Data is split into Test (Happy - 200 / Not Happy - 224) and Train (Happy - 2000/ Not Happy - 2022) data set

Sample Images



Sample Images

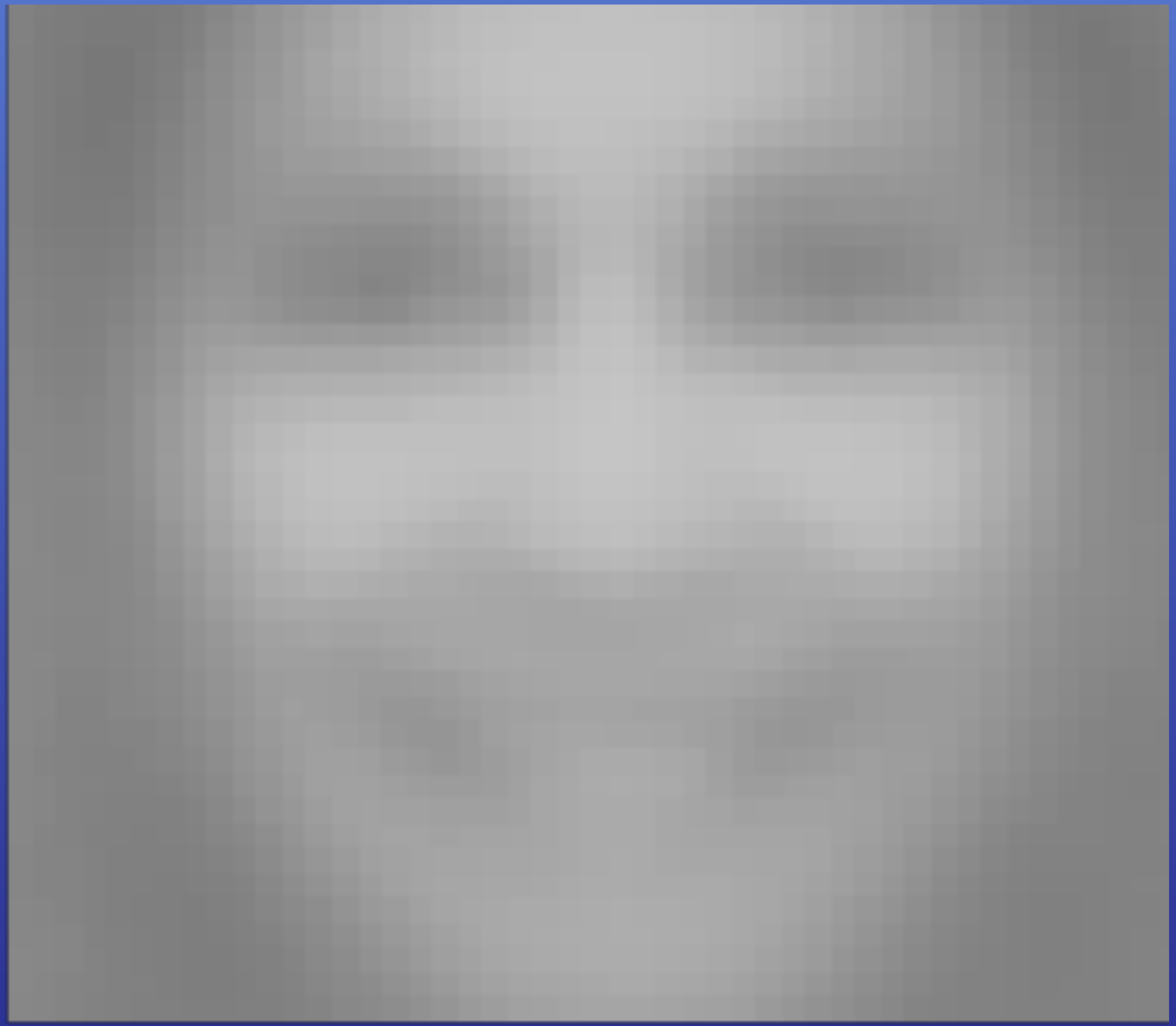


Mean and std for Happy images:

Count: 2000

Mean: 130.58182

Standard Deviation: 63.10906

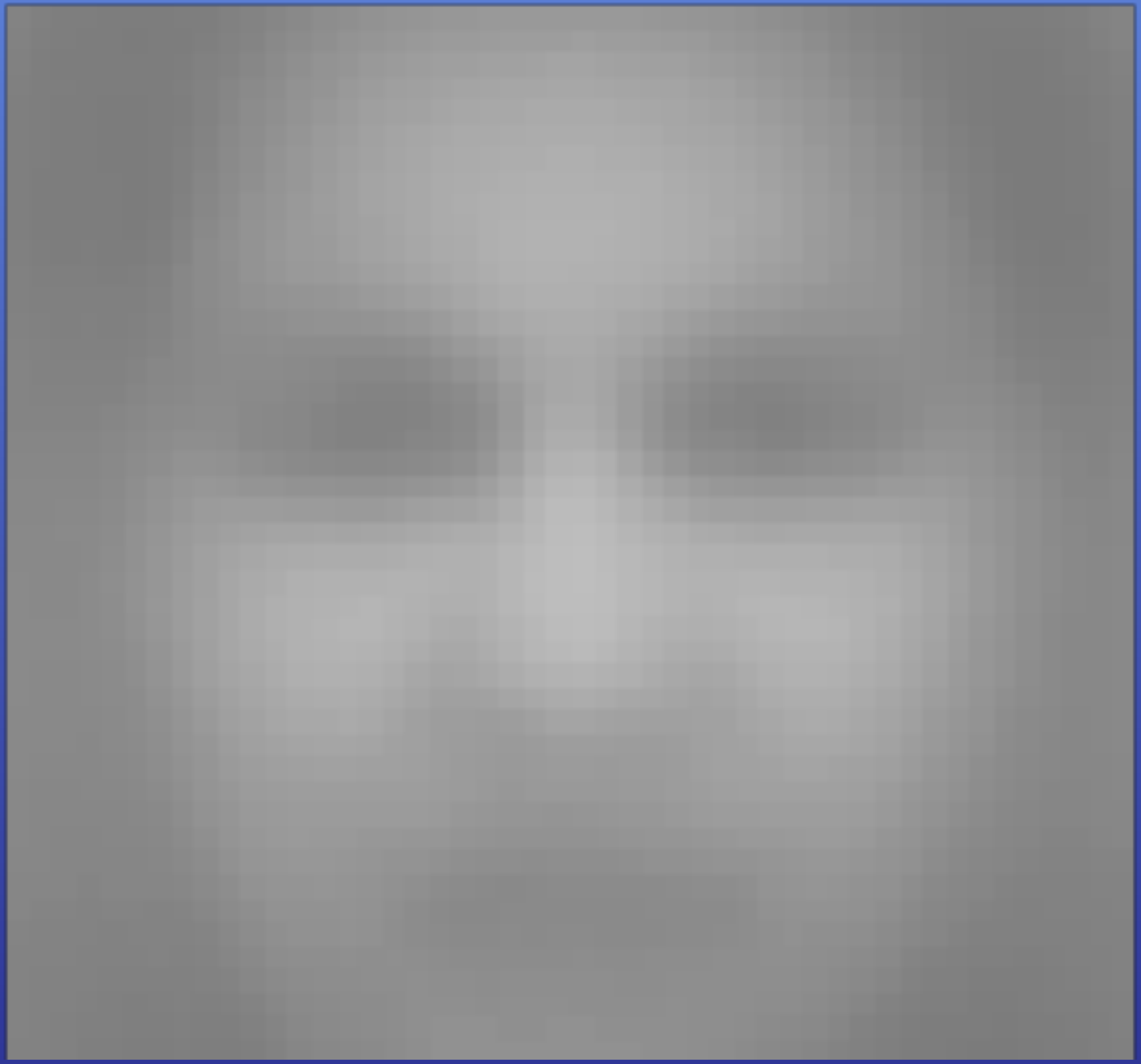


Mean and std for Not_Happy images:

Count: 2022

Mean: 124.172104

Standard Deviation: 65.34936



Data pre-processing

The goal of pre-processing is to prepare the raw data in a format that is suitable for training a model.

Argumentation Techniques can be applied to Training data to improve model performance

Flip and Rotation help model to perform well in real life images orientation which can be upside down.

Blurring can reduce noise and focus on key feature

Standardize the data

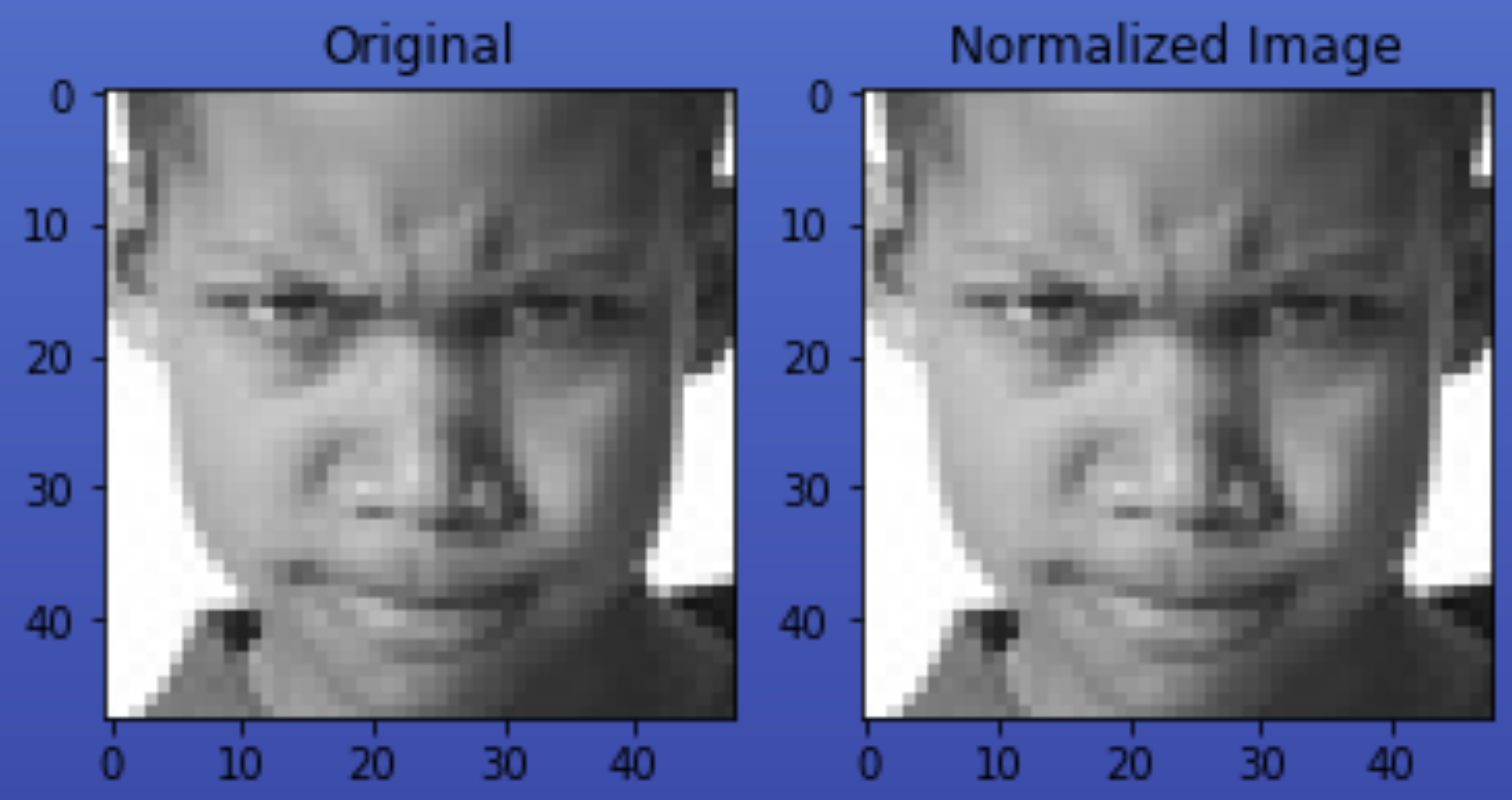
Resizing

- Resize images to a standard size. CNNs often require input images to have consistent dimensions. You can choose a size that balances computational efficiency and information retention.

Normalization:

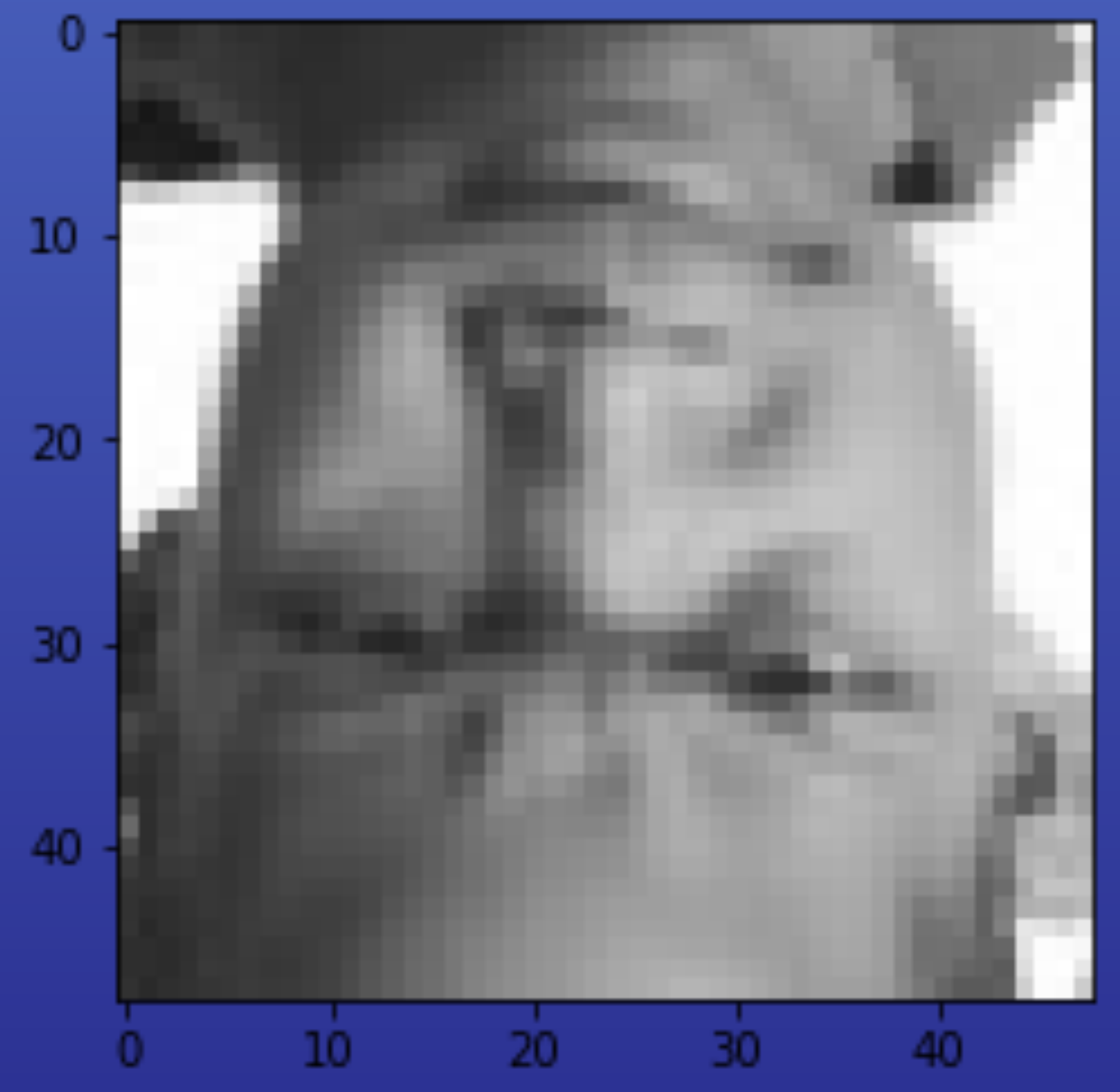
- Normalize pixel values to a standard range, such as [0, 1] or [-1, 1]. This hel faster convergence during training.

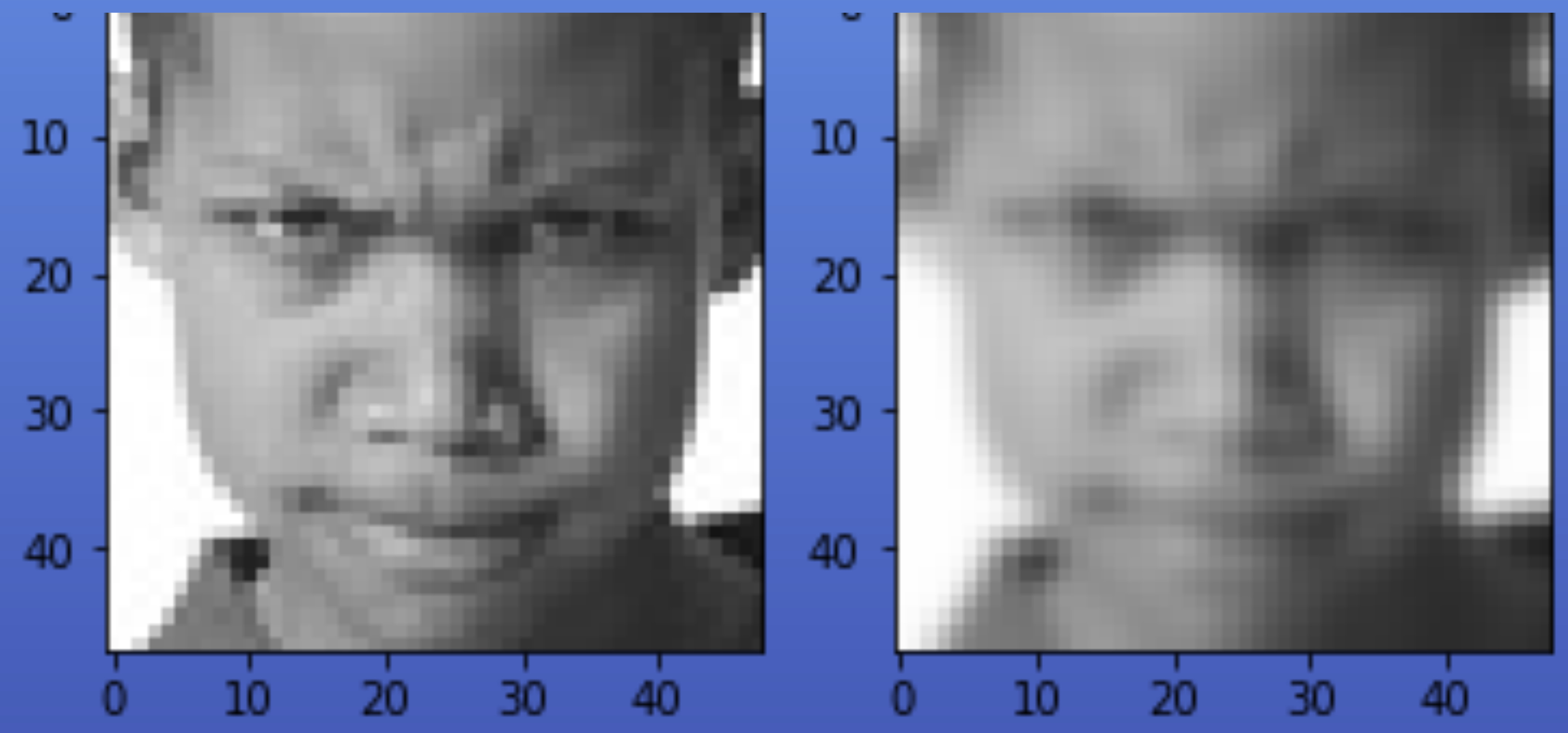
Standardize the data



Argumentation Techniques

Augment the dataset to increase its diversity and improve generalization. Common techniques include rotation, scaling, flipping, and changes in brightness and contrast





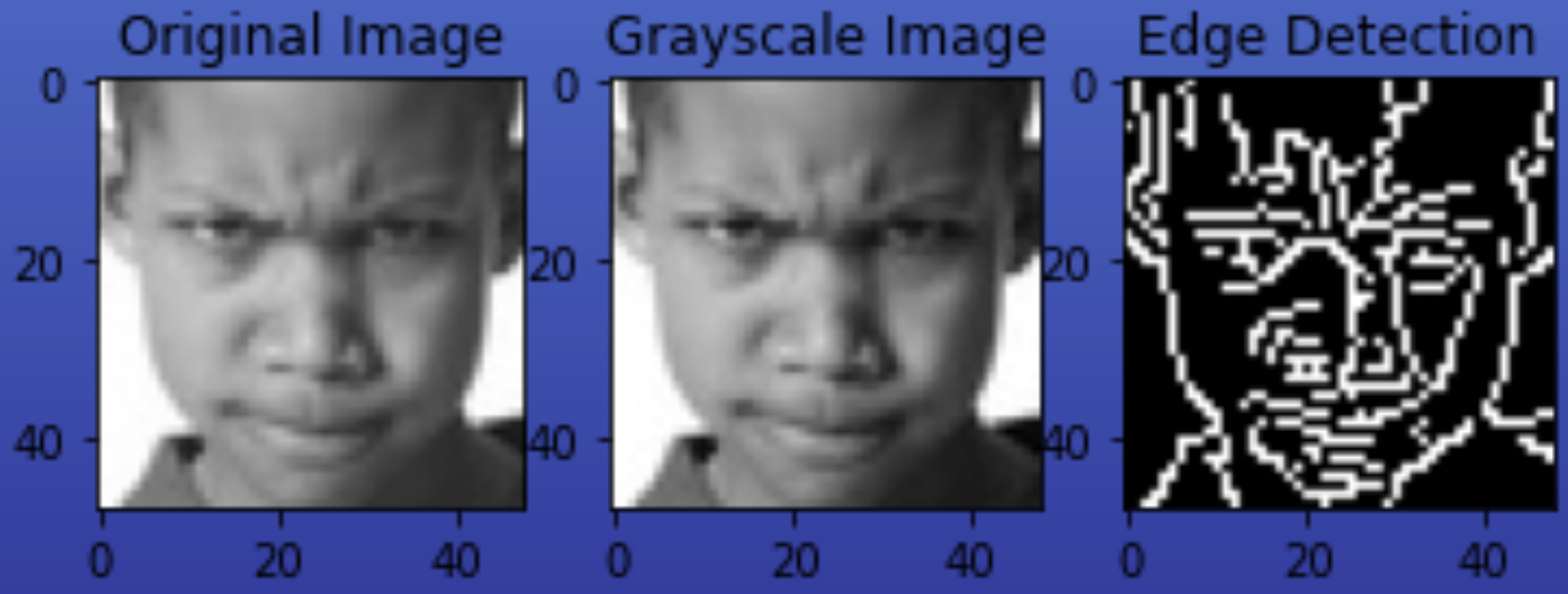

Original (BGR)
RGB
HSV

Grayscale Conversion:

Convert images to grayscale if color information is not crucial for your task. This reduces computational complexity and can be beneficial if color is not a significant factor in emotional expression

Noise Reduction:

Apply techniques like blurring or denoising to reduce noise in images, which can improve the model's ability to focus on relevant features.


Original Image
Grayscale Image
Edge Detection
0
20
40
0
20
40

Modeling Process

Grayscale Conversion:

Convert images to grayscale if color information is not crucial for your task. This reduces computational complexity and can be beneficial if color is not a significant factor in emotional expression

Noise Reduction:

Apply techniques like blurring or denoising to reduce noise in images, which can improve the model's ability to focus on relevant features.

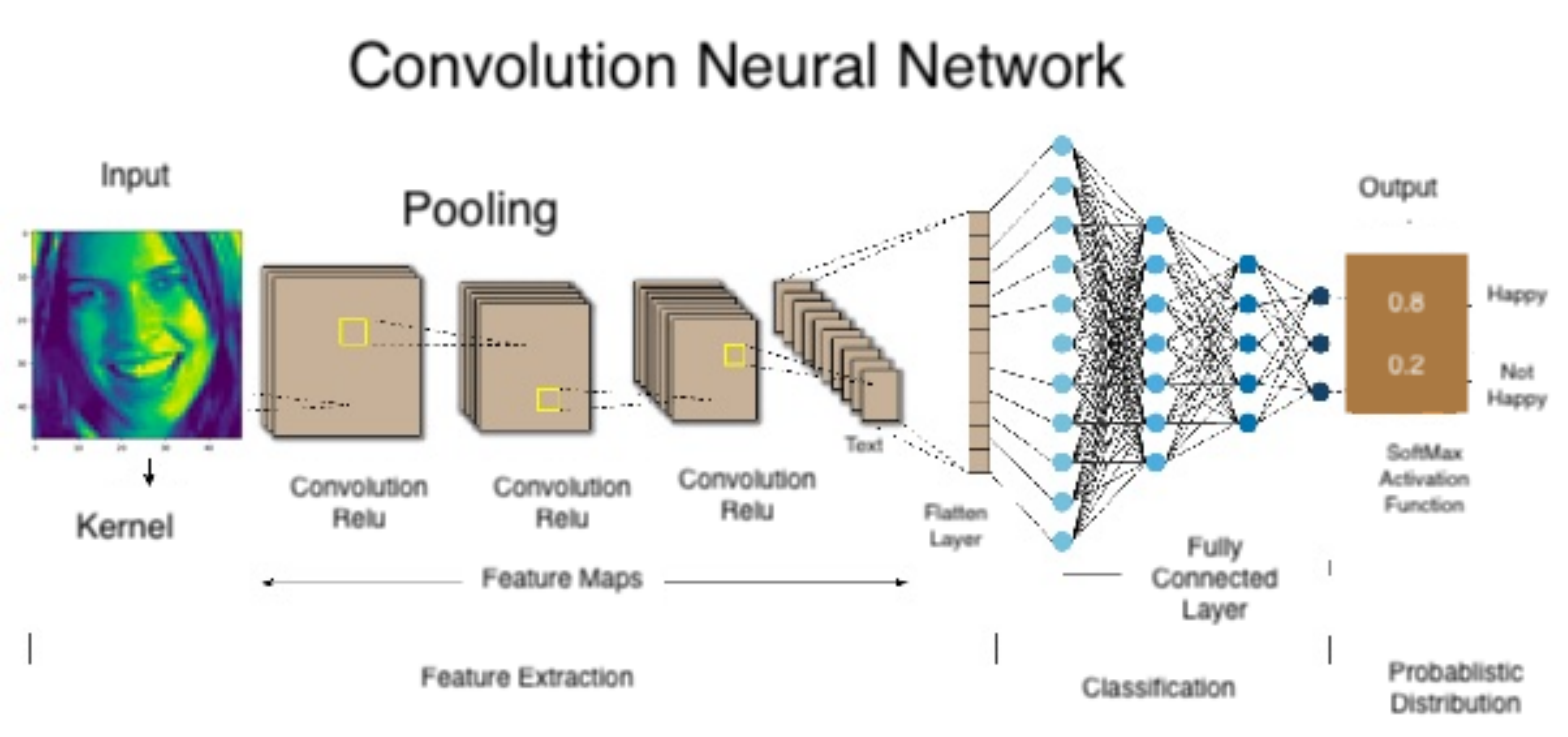
Base CNN Model

Convolution Neural Network: Specialized for image and grid-like data.

Extract Feature - Convolution is the process of adding each element of the image to its local neighbors, weighted by the kernel.

Classification - Convert Convolution Matrix to flatten input for fully connected dense neural network and applying softmax to classify the output

Base CNN Model



Base CNN Model

Label Encoding:

- Encode emotion labels into numerical format.

  'Happy': 0 , 'Not_Happy': 1

Base CNN Model

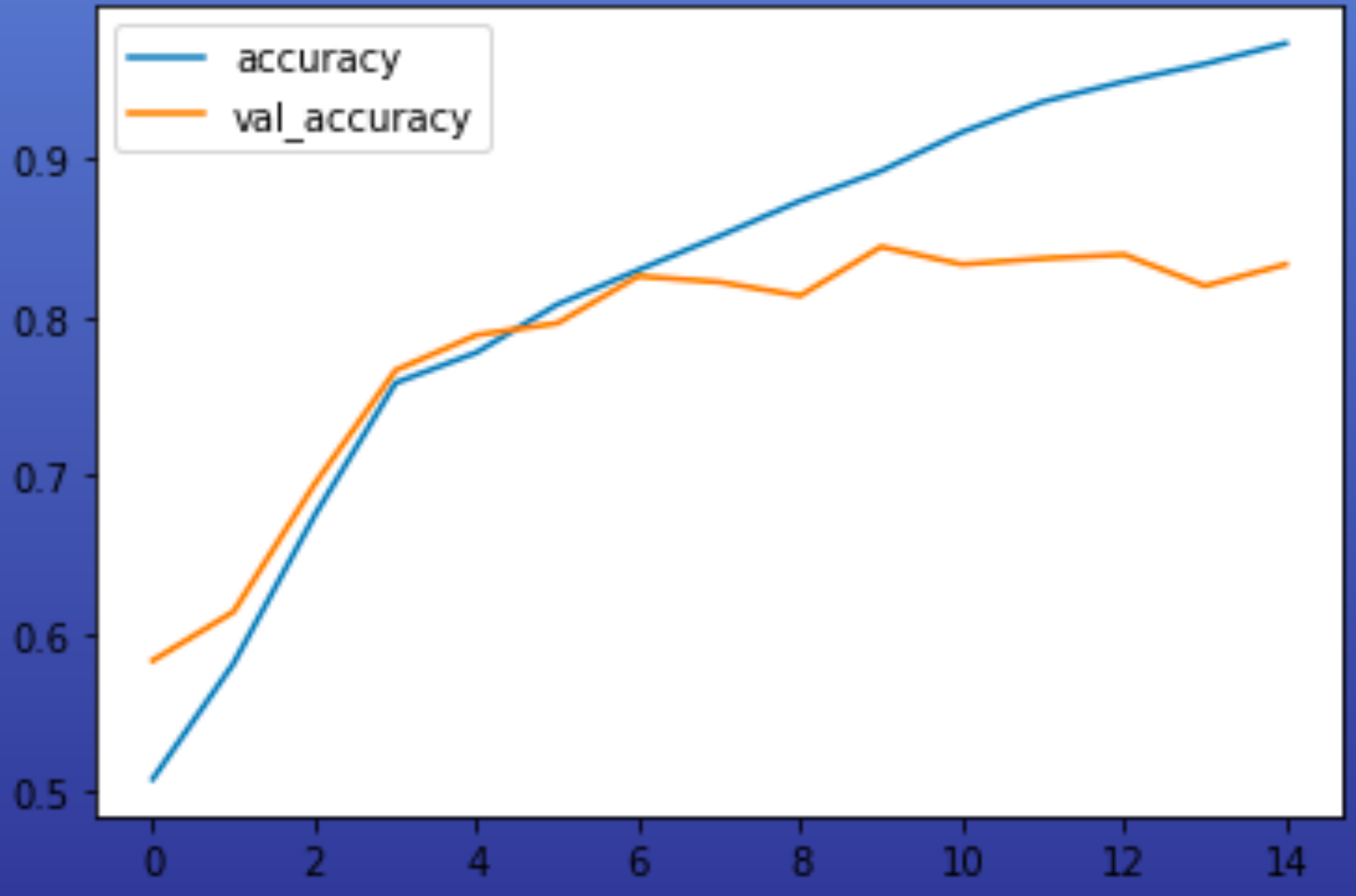
```
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 46, 46, 32)        320

 max_pooling2d (MaxPooling2  (None, 23, 23, 32)        0
 D)

 conv2d_1 (Conv2D)           (None, 21, 21, 64)        18496

 max_pooling2d_1 (MaxPoolin  (None, 10, 10, 64)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 8, 8, 128)         73856

 flatten (Flatten)           (None, 8192)              0

 dense (Dense)               (None, 112)               917616

 flatten_1 (Flatten)         (None, 112)               0

 dense_2 (Dense)             (None, 112)               12656

 dense_3 (Dense)             (None, 2)                 226
...
Total params: 1023170 (3.90 MB)
Trainable params: 1023170 (3.90 MB)
Non-trainable params: 0 (0.00 Byte)
```

Base CNN Model

Base Model performance is good on training set.
loss: 0.0704 - accuracy: 0.9736 (overfitting)

But it doesn’t perform good on validation set
val_loss: 0.6828 - val_accuracy: 0.8335

Base CNN Model



Base CNN Model

Base Model performance is good on training set.
loss: 0.0704 - accuracy: 0.9736 (overfitting)

But it doesn’t perform good on validation set
val_loss: 0.6828 - val_accuracy: 0.8335

Creating Tuner -Keras Tuner

Keras Tuner makes it easy to define a search space and leverage included algorithms to find the best hyperparameter values using random search algorithm.

Tuner best fitted model summary -

Creating Tuner -Keras Tuner

Tuner best fitted model summary -

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 random_flip (RandomFlip)    (None, 48, 48, 1)         0

 random_contrast (RandomCon  (None, 48, 48, 1)         0
 trast)

 random_rotation (RandomRot  (None, 48, 48, 1)         0
 ation)

 conv2d (Conv2D)             (None, 44, 44, 48)        1248

 conv2d_1 (Conv2D)           (None, 42, 42, 416)       180128

 conv2d_2 (Conv2D)           (None, 38, 38, 192)       1996992

 flatten (Flatten)           (None, 277248)            0

 dense (Dense)               (None, 80)                22179920

 dense_1 (Dense)             (None, 2)                 162

=================================================================
...
Total params: 24358450 (92.92 MB)
Trainable params: 24358450 (92.92 MB)
Non-trainable params: 0 (0.00 Byte)
```

Tune Model Performance

Using Tensor Flow Tuner module we can find optimize layers, learning rate and many other convolution network configuration

Applied preprocessing random rotation , random flipping and random contrast to eliminate bias and add more sample.

Based on Tensor flow Tuner . Best fit model for

CNN is Layer contains 3 layers

Total Trained Parameters - 24358450
Non Trainable parameter - 0

Tunned CNN Model

The model performance good on Training dataset with

True Postive : 87%
Precision - 78% and F1 - 87% Accuracy - 83%

	precision	recall	f1-score	support
0	0.78	0.87	0.82	200
1	0.87	0.79	0.83	224
accuracy			0.83	424
macro avg	0.83	0.83	0.83	424
weighted avg	0.83	0.83	0.83	424

Recommendation

The model performance is good on Training set. It can be deployed and integrated on production environment. Model needs to be continued to be monitored and improved.

For hospitality industry to be successful and profitable, it is important to evaluate customer experience. This model can be deployed in production to analyze customer facial expression along with feedback on checkout. Based on the model prediction one can evaluate performance and service provided. Improve customer experience by providing customized service based on the model.