```
[Start]
                            [Install and Import Libraries]
        (numpy, pandas, tensorflow, cv2, matplotlib, sklearn, webbrowser)
                                 [Define Constants]
                  (emotion_classes, recommendations dictionary)
                              [Load FER2013 Dataset]
                                 (Read fer2013.csv)
                                   [Split Dataset]
(Training: Usage='Training', Validation: Usage='PublicTest', Test: Usage='PrivateTest')
                                 [Preprocess Data]
(Handle missing values, convert pixels to arrays, normalize, one-hot encode labels)
               [Output: X_train, y_train, X_val, y_val, X_test, y_test]
                                 [Build CNN Model]
      (Sequential: Conv2D, BatchNorm, MaxPooling, Flatten, Dense, Dropout)
                                 [Compile Model]
       (Optimizer: adam, Loss: categorical_crossentropy, Metrics: accuracy)
                                   [Train Model]
  (Fit with X_train, y_train, validate with X_val, y_val, epochs=10, batch_size=64)
                                 [Evaluate Model]
                     (Test on X_test, y_test, print test accuracy)
                                         \mathbf{T}
```

```
[Save Model]
                           (Save as emotion_recognition_model.h5)
                                 [Prompt User for Input Mode]
                                     (Enter 'image' or 'live')
                                 [Decision: Input Mode Valid?]
                             Z
                    [Valid]
                                                                     [Invalid]
            [Decision: Mode?]
                                                            [Print "Invalid mode selected"]
              Z
                                  V
         [Image]
                                       [Live]
                                                                             [End]
    [Image Mode]
                                       [Live Mode]
[Prompt for Image Path]
                                        [Initialize Webcam]
[Load & Preprocess Image]
                                      [Capture Frame]
(Read, grayscale, resize, normalize) (Convert to grayscale)
[Predict Emotion]
                                          [Detect Faces]
 (Use model.predict)
                                          (Haar Cascade)
[Display Image with Emotion]
                                         [For Each Face]
 (Show with matplotlib)
                                      (Extract ROI, resize, normalize)
[Store Predicted Emotion]
                                         [Predict Emotion]
                                        (Use model.predict)
                                           [Annotate Frame]
```

```
(Draw rectangle, label emotion)
                              [Display Frame]
                         (Show with cv2.imshow)
                             [Decision: Press 'q'?]
                                  [Yes] [No]
                                     K
                        [Release Webcam, Close Windows]
                        [Store Last Predicted Emotion]
V
                                            Z
                  [Provide Recommendations]
           (Retrieve from recommendations dictionary)
                 [Print Quote, Activity, Playlists]
                    (English and Telugu songs)
                   [Prompt for Playlist Choice]
                    (Enter 'English' or 'Telugu')
                    [Decision: Playlist Valid?]
                                           7
                 [Valid]
                                         [Invalid]
         [Open First Song URL] [Print "Invalid choice"]
        (Using webbrowser.open)
                \mathbf{\downarrow}
                                             [End]
             [End]
```