Mimic Me Emoji game using Affectiva API

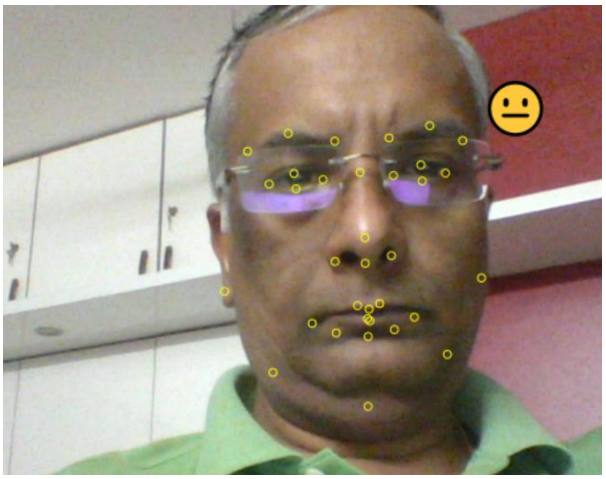
This exercise required use of webcam to capture a face image and then Affectiva API to extract the most dominant emoji.

Implementation Details

Task 1 – Display Feature Points

The task was about completing the code in function “drawFeaturePoints”. Steps involved for the same were:

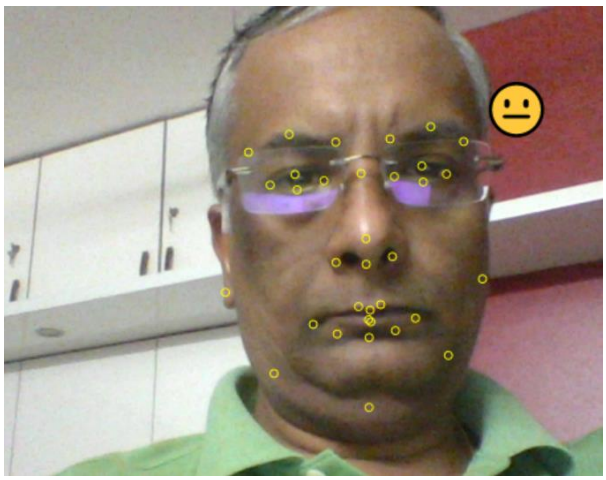
a) Get 2D context from the canvas passed to function as an argument
b) Set the style for point markers. I set the stroke color to yellow to draw yellow colored circles.
c) Loop over all the feature points in the “face” object passed to the function as an argument.
    a. Extract a feature point
    b. Draw a yellow colored circle at the feature point coordinates. Circle was drawn using ctx.arc() function with angle range(0, 2*pi)



Task 2 – Show Dominant Emoji

The task was about drawing the dominant emoji close to the face and vary the size of emoji based on the size of face detected: This required completion of function “drawEmoji”. Steps involved were:

a) Get 2D context from the canvas passed to function as an argument.
b) Iterate over all points to:
    a. Get x_min, x_max, y_min
    b. Scale the emoji based on x-axis spread (x_max-x_min)
    c. Draw emoji at coord (x_max, y_min) i.e. towards right upper side of the face
    d. Emoji is drawn using ctx.fillText() function



Task 3 – Implement Mimic Me!

The task was to implement a basic game where a random emoji is shown on the screen and user is expected to mimic the emoji. User needs to mimic 5 times to complete the game. Time taken to complete the challenge is then shown to the user. Steps involved were:

a) Create global variables
    a. cur_score: current score – initialized to zero
    b. total_score: total score – initialized to 5
    c. Load audios to play when user’s emoji matches the target and another when the game is completed
    d. emoji_to_mimic: the current emoji that user needs to mimic
b) inside “onInitializeSuccess” function, do:
    a. set cur_score to zero – score at the start of game
    b. pick a target emoji at random
    c. show the target emoji using “setTargetEmoji()’ function
    d. show the score using “setScore()” function

c) Inside “onImageResultsSuccess” function, do:
    a. Check if the dominant emoji from user’s image is same as screen and if yes,

i. increase the current score by one.
ii. Pick a new target emoji
iii. If the score has reached total_score then freeze the game and play “game end cheers” sound. Also give a visual indication next to score to let the user know the time it took for him/her to complete the game.

d) Inside “onReset” function, initialize the scores similar to that for “onInitializeSuccess” in (b) above.