Project: Mimic Me!

Playing From Your browser

The demonstration application including the debug information from Affectiva's API calls is seen here;

Demonstration

For a truly special version visit: NYAN Version (Not working on mobile browsers...yet)

m key mutes the music, you're welcome

Overview

This project tracks the faces in a video and identify facial expressions using Affectiva API. We are able to match the expression to a fixed set of emoji and compare them in real-time.

Getting Started

We'll be using Affectiva's Emotion-as-a-Service API for this project. Visit their Developer Portal and try out some of the sample apps. Affectiva makes it really easy to extract detailed information about faces in an image or video stream. To get a sense for what information you can obtain, check out the Metrics page.

Serving locally over HTTPS

In order to access the webcam stream, modern browsers require you to serve your web app over HTTPS. To run locally, you will need to general an SSL certificate (this is a one-time step):

- Open a terminal or command-prompt, and ensure you are inside the `AIND-CV-Mimic/` directory.
- Run the following shell script: `generate-pemfile.sh`

This creates an SSL certificate file named `my-ssl-cert.pem` that is used to serve over https.

Now you can launch the server using:

```
python serve.py
```

Note: The `serve.py` *script uses Python 3.*

Running and implementing the game

Open a web browser and go to: https://localhost:4443/

- Hit the Start button to initiate face tracking. You may have to give permission for the app to access

your webcam.
- Hit the Stop button to stop tracking and Reset to reset the detector (in case it becomes stuck or unstable).
- When you're done, you can shutdown the server by pressing `Ctrl+C` at the terminal.

Note: Your browser may notify you that your connection is not secure - that is because the SSL certificate you just created is not signed by an SSL Certificate Authority. This is okay, because we are using it only as a workaround to access the webcam. You can suppress the warning or choose "Proceed Anyway" to open the page.

Background Tasks

1. Display Feature Points

The API returns an array of feature points from the service. These represent key points on the face. The code uses the HTML5 canvas object and draws a small circle at the point location.

```
// Draw the detected facial feature points on the image
function drawFeaturePoints(canvas, img, face) {
  // Obtain a 2D context object to draw on the canvas
  var ctx = canvas.getContext('2d');

  // Loop over each feature point in the face
  for (var id in face.featurePoints) {
    var featurePoint = face.featurePoints[id];
      drawPoint(featurePoint.x, featurePoint.y,ctx);
  }
}
function drawPoint(x, y, canvas){
  canvas.beginPath();
  canvas.arc(x, y, 1, 0, 2 * Math.PI, true);
  canvas.stroke();
}
```

2. Show Dominant Emoji

In addition to feature points and metrics that capture facial expressions and emotions, the Affectiva API also reports back what emoji best represents the current emotional state of a face. This is referred to as the *dominant emoji*.

The emoji is drawn using a HTML5 fillText call. We fix the size at 48px but track the emoji to the users face. Trial and error found that point[13] in the list was typically the center of the face.

```
// Draw the dominant emoji on the image
function drawEmoji(canvas, img, face) {
  // Obtain a 2D context object to draw on the canvas
  var ctx = canvas.getContext('2d');

  var emoji = face.emojis.dominantEmoji;
```

```
    // Usually its the center
    var featurePoint = face.featurePoints[13];

    if(emoji != 0){
      ctx.font = '48px serif';
      ctx.fillText(emoji, featurePoint.x, featurePoint.y);
    }
  }
```

3. Game

Taking the same metric and adding some game play mechanism you can see how Affectiva works here; NYAN Version

Mobile browser support is not supported on the Affectiva SDK yet

Affectiva Resources

Refer to resources in Affectiva's JS SDK documentation.

Other references:

- Affectiva Developer portal
- Demo that this project is based on
- Tutorials: Camera stream, video, photo