BALAN Anastasiya
THOUVENIN Estelle

BlendedFaces

Introduction

BlendedFaces is a facial recognition project created by Anastasiya Balan and Estelle Thouvenin, and based on ml5's Facemesh. You can access our github project thanks to this link : https://github.com/EstelleThvn/ea_creative_project. You can also see the final result here : https://estellethvn.github.io/ea_creative_project/.

To use this feature you need to allow the use of your webcam, then wait for the video to be loaded. When the video will be loaded you must see yourself on the left side of the screen. Then you must click on the button "choisir un fichier", where you have to upload a face image. It will work better if the face image uploaded is clear and is from the front. Then, you will be able to see facial recognition points on your face and your mouth and eyes on the face image you uploaded. Note that the first image loading might take a bit of time.

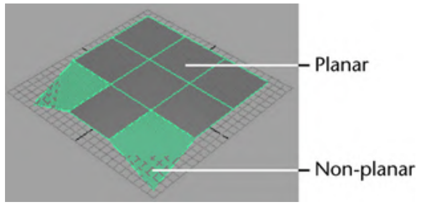
1.From the first idea to the documentation

1.1. What did we want to work on ?

We were really interested in working with facial 3D coordinates. And ml5 offered us 2 ways to work on that : Facemesh and Face API. Let's see what they propose :

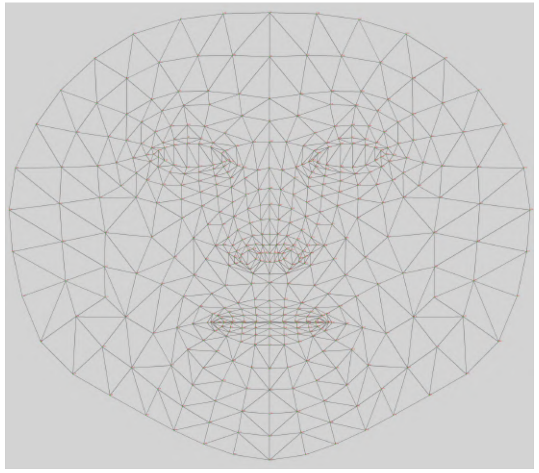
- **Facemesh**

Facemesh is a machine-learning model that allows the detection of facial landmarks in the browser. There are 486 3D points that describe the geometry of each face. To be precise with the 3D face recognition, we are working with triangles. Triangles are more precise especially in complex areas such as the nose, the eyes and the lips. Triangles are very used in 3D modelling, but why ? Why not use quads or polygons ?



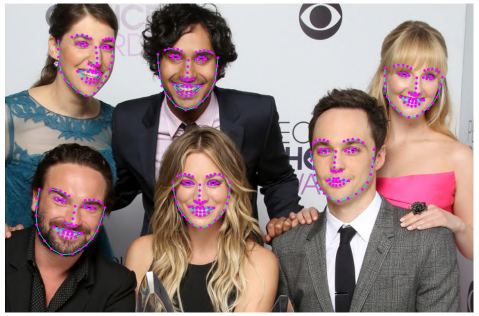
Triangles can never be non-planar; anything with more than 3 points can be non-planar and thus un-renderable unless converted to triangles. Also, using triangles is memory efficient because they can be sorted, and rendered extremely fast.

Here is the face model Facemesh is using :



- **Face API**
  Face API allows you to access face and face landmark detection too. It is used for Face Recognition, Face Landmark Detection, Face Expression Recognition, and Age Estimation and Gender Recognition.

We got more interested in the use of face Landmark (as you'll see in the scenario section). Here is an example :



We notice that face API is using much less points and is, thus, less precise than Facemesh. We decided to use Facemesh for this reason.

1.2. For what user scenario ?

First we were thinking about using this technology to recreate the deep fake mechanism. As we can localize moving faces, we could move the points' coordinates of a static image according to the moving face. But that would take more time and raises questions such as: what if the uploaded face image has a closed mouth but the facial recognition points detected an open mouth ?

The second idea was the instagram filter which consists in adding an image to specific points (a red circle on the center of the nose, sunglasses on the eyes...). But that is also raising some problems: as we are putting a 2D image on a 3D video, what happens when the person turns their head ?

So we kept our last scenario which is a mix of the first and the second one. It is more likely to look like a face swap (cf Snapchat filter), and therefore be used as an entertaining feature. But if we had a bit more time we could try to create a pseudo deep fake. As you'll see in a few paragraphs, we copied the content of specific areas (with masks) and pasted it on the loaded image, 25 times a second. So yes this project was created for entertaining purposes, as for example in the case of a filter in a social media such as Instagram or Snapchat, because this is obviously funny and made us both laugh a lot while coding. But it is a light way to sensitize people on deepfakes. Yes it is funny, but what if we deleted all the funny parts (zooms in the eyes) and created a more realistic look ? This could also be used in an educational purpose because it would sensitize people on this issue.

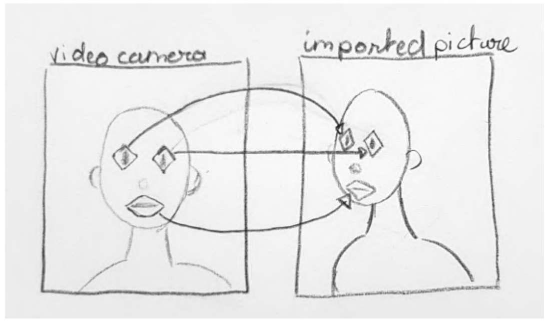
1.3. Looking for inspirations

Facial recognition or deepfake technologies really inspired us : it was the seeds of our project. Then it evolved next to social media filters such as face swap, flower filters or animal filters.

2.From first visuals to a strong concept

2.1 The outlines and sketches of our idea

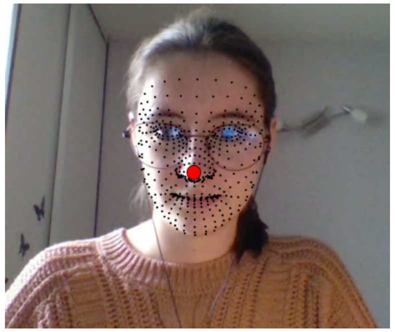
First we needed to know what we were trying to achieve. In order to take elements from the video and put them on another picture, we had to copy those elements according to some precise points on the face (given by Facemesh), and paste them onto the other picture.

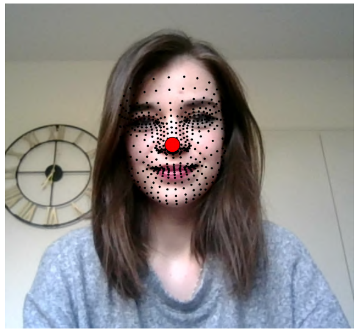


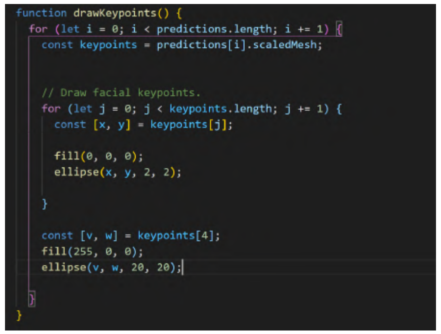
To achieve that, we created a pseudocode :

- get the position of each point of the face mesh from the webcam footage (25 images per second),
- create the shapes for the eyes and the mouth from those points,
- copy the image from the video camera within those shapes,
- get the position of each point of the face mesh from the imported picture (imported by the user),
- get the position of the eyes and the mouth from the imported picture,
- put the copied images of the eyes and mouth on top of the eyes and mouth on the imported picture.

Then, we started using Machine Learning thanks to Facemesh and tried to find specific points on the face captured by the webcam. This function creates the points on the face for each image from the video and highlights the chosen specific point.







```
function drawKeypoints() {
  for (let i = 0; i < predictions.length; i += 1) {
    const keypoints = predictions[i].scaledMesh;


    // Draw facial keypoints.
    for (let j = 0; j < keypoints.length; j += 1) {
      const [x, y] = keypoints[j];

      fill(0, 0, 0);
      ellipse(x, y, 2, 2);

    }

    const [v, w] = keypoints[4];
    fill(255, 0, 0);
    ellipse(v, w, 20, 20);

  }
}
```

2.2. A strong entertaining concept

BlendedFaces is THE trendy filter. Select an image of one of your friends or a celebrity, one of your choice, and let the site put your mouth and eyes over the selected image. Make them do weird faces, sing the worst songs or make them say anything you want.

3.One of the funniest filters

3.1 The code behind the filter

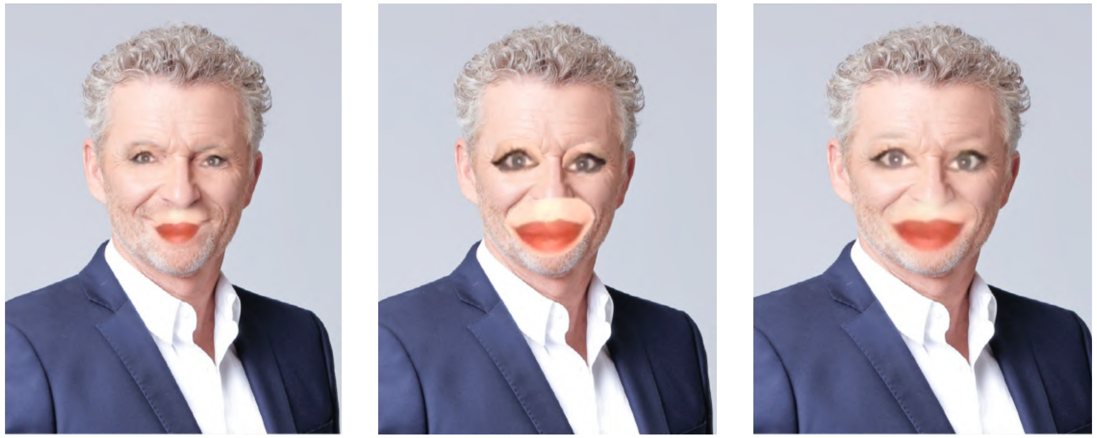
To make our project more entertaining and funny, we scale up the eyes and the mouth from the webcam video when we display them onto the imported picture.

```
image(picsMouth[idx], width/2+xMouthUserImg+widthMouthUserImg/2-widthMouthDeepfake/2*1.5, yMouthUserImg+heightMouthUserImg/
2-heightMouthDeepfake/2*1.5, widthMouthDeepfake*1.5, heightMouthDeepfake*1.5);
```
(line 170 on sketch.js)

We also decided to add a blur effect to the mask that contains the copied eyes and mouth to have them blend more with the imported picture. It adds a bit of credibility to the result.

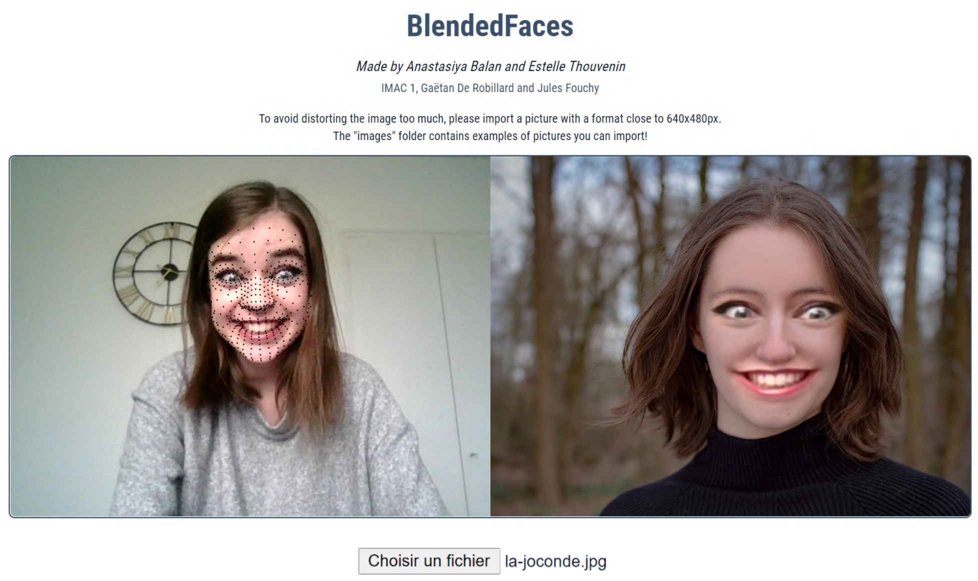
```
let img = video.get(keypoints[57][0]-10, keypoints[164][1]-10, widthMouth+20, heightMouth+20);
```
(line 127 on sketch.js)

```
mouthMask.filter(BLUR, 8);
img.mask(mouthMask);
```
(line 160-162 on sketch.js)


(without scale and with blur / with scale and without blur / with scale and with blur)

3.2 The final result

And here is the outcome of BlendedFaces!



BlendedFaces

Made by Anastasiya Balan and Estelle Thouvenin

IMAC 1, Gaëtan De Robillard and Jules Fouchy

To avoid distorting the image too much, please import a picture with a format close to 640x480px.
The "images" folder contains examples of pictures you can import!

Conclusion

BlendedFaces is a funny experience we created thanks to ml5's Facemesh. However, this experience can be closely related to deepfakes : at which point do we go from entertainment to danger? In the meantime, have fun using this filter and don't forget to share it in the discord channel "esthétique-algorithmique"!

Bibliography

https://learn.ml5js.org/#/reference/facemesh
https://editor.p5js.org/ml5/sketches/Facemesh_Webcam
https://arxiv.org/pdf/1907.06724.pdf
https://github.com/tensorflow/tfjs-models/tree/master/facemesh