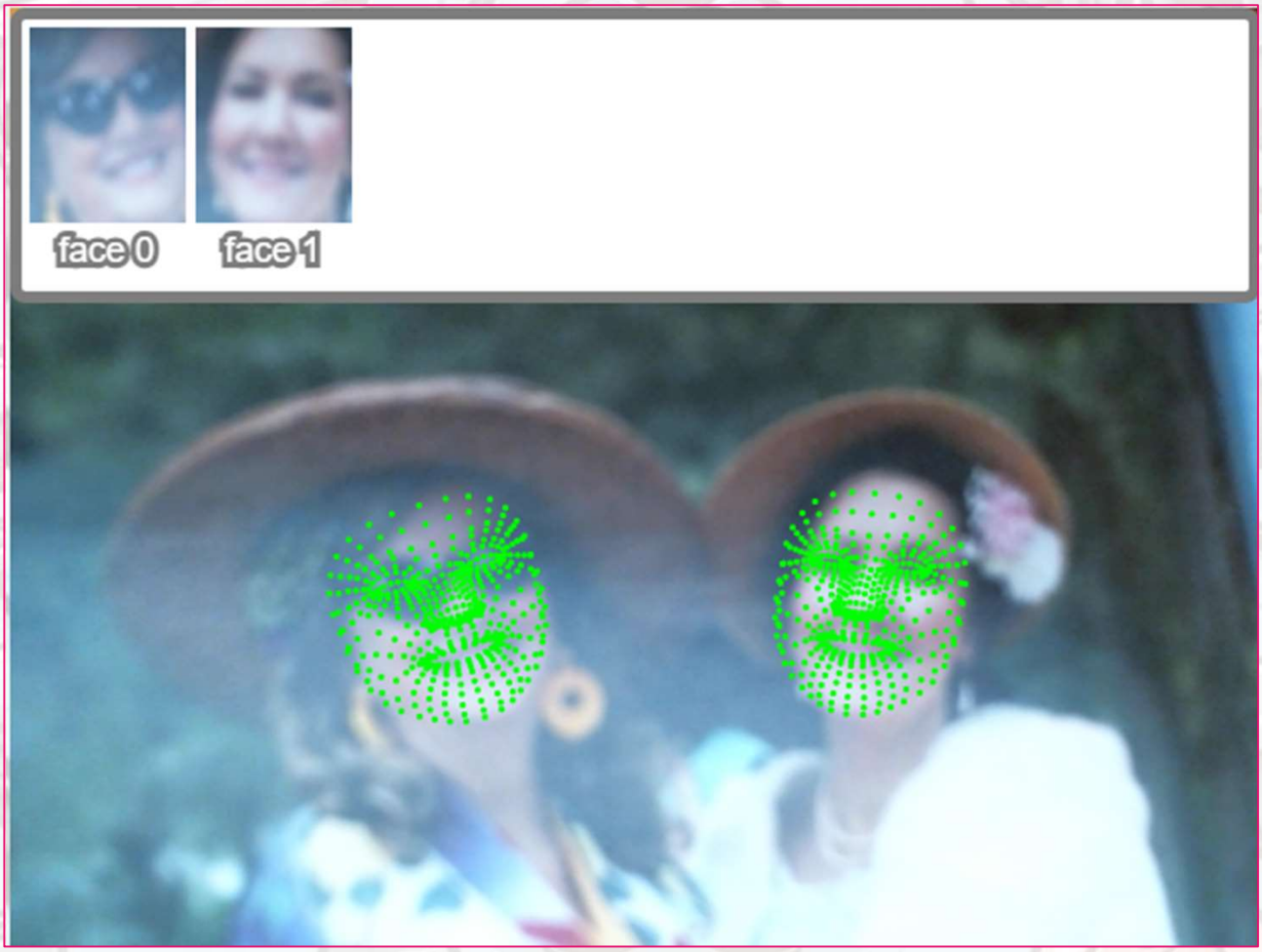
AI Face Tracking

University of Sunderland

In this activity you will build an application that uses Google's **`MediaPipe`** AI Models to recognise faces from a webcam feed.



The basic idea is you will incrementally build a web browser-based application that integrates a webcam and the **`FaceMesh`** model (a part of **`MediaPipe`**) to create a Facial recognition surveillance tool.

Let’s get going!

University of Sunderland

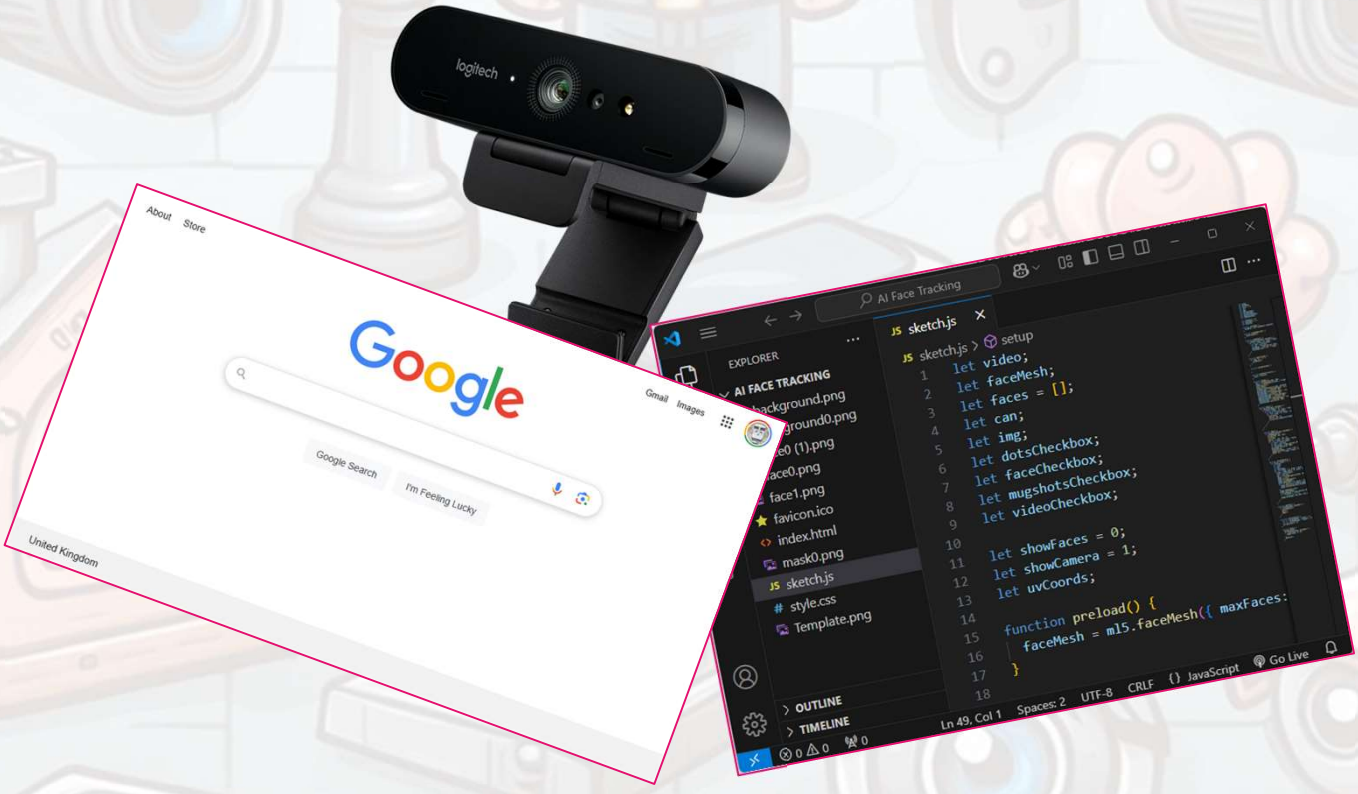
What do we need!

For this activity we need the following:

- A **web browser** (we recommend you use Chrome).
- The **VS Code** Editor
- A **webcam** connected to the computer.
- **Your Face (and others ☺)!**

The lab computers should already be setup, but if you get stuck at any point, ask for help. We are here to help you. Don't worry if you don't get everything finished in this session there are sample masks in the application you download already.

Source files: https://github.com/neliot/FaceMesh



Let's get going!

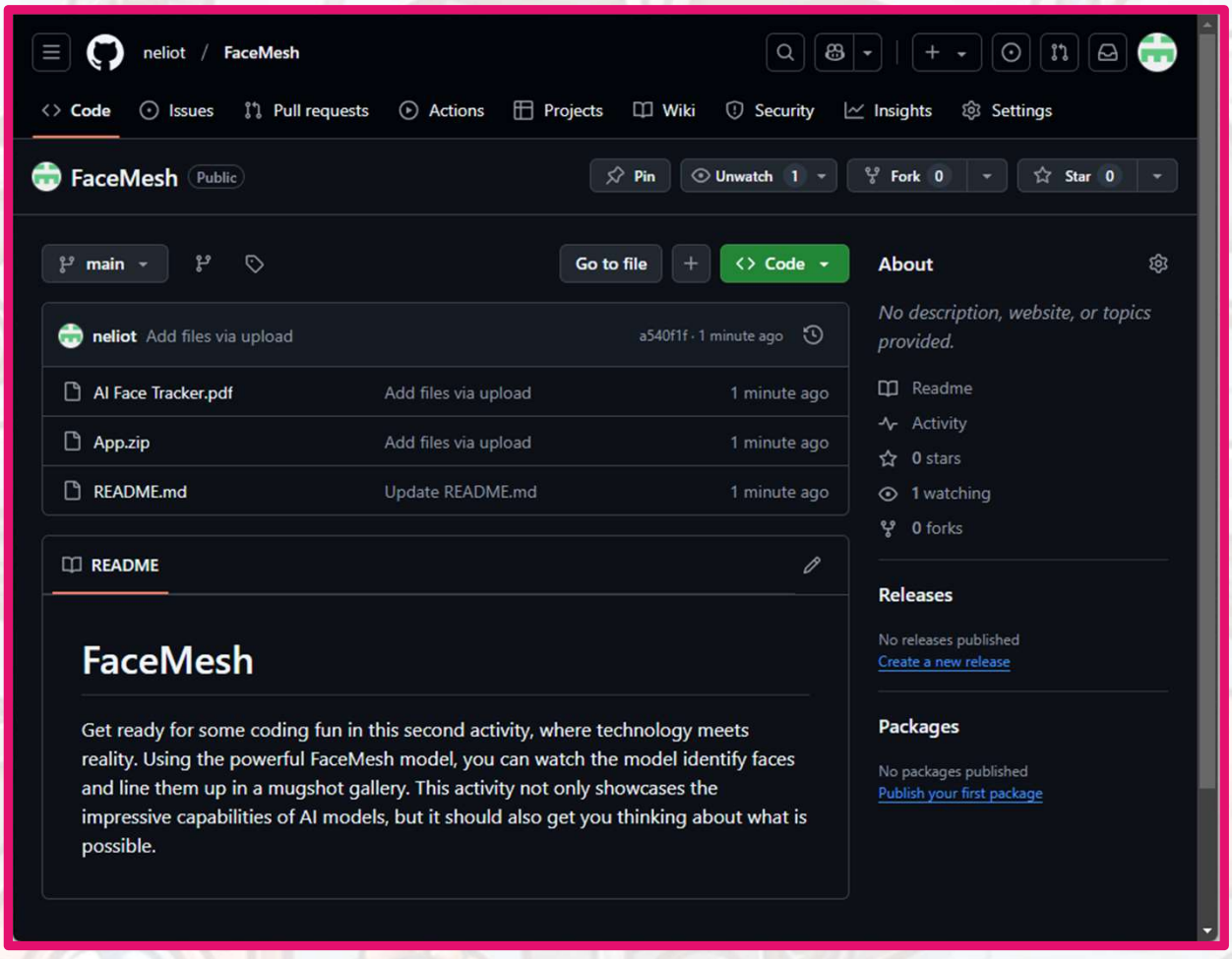
University of Sunderland

Download the application!

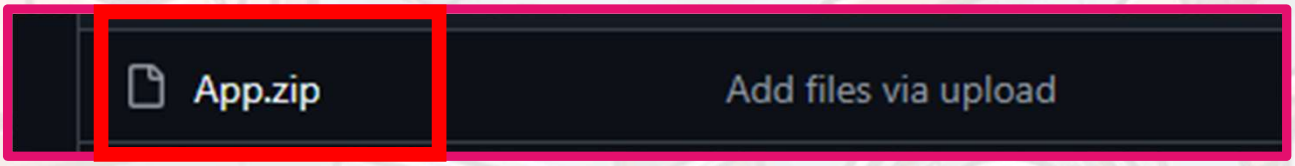
In your browser enter the URL below:

https://github.com/neliot/FaceMesh

You should see a page similar to the one below:



Click on the App.zip file:

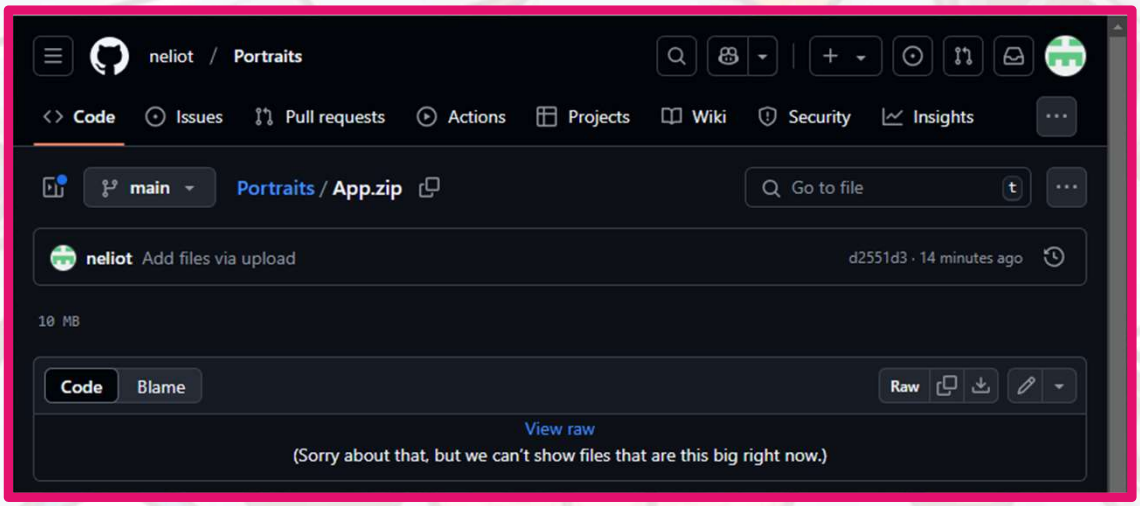


Let's get going!

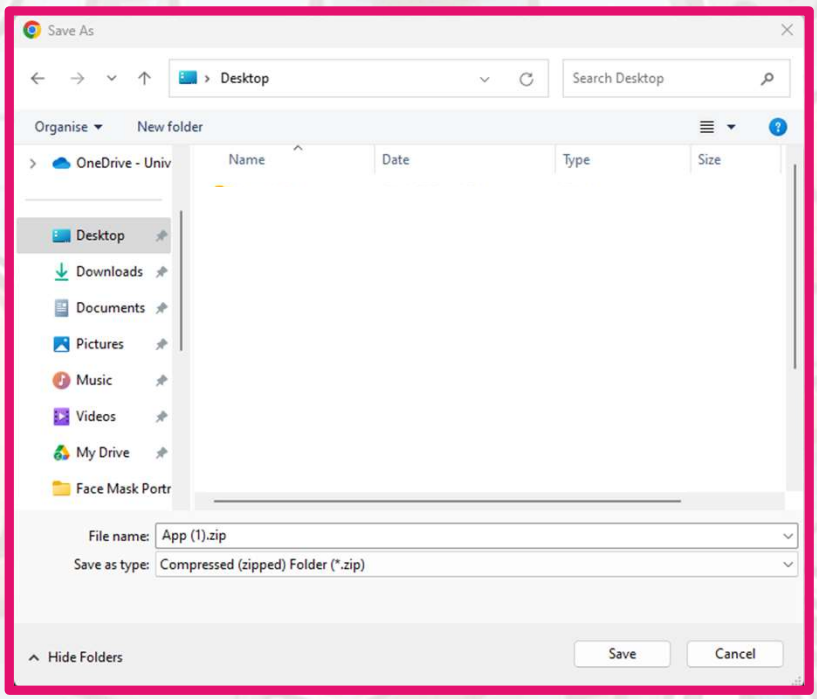
University of Sunderland

Download the application!

You will now be presented with the following screen:



Click on the download icon and save the file to the Desktop.

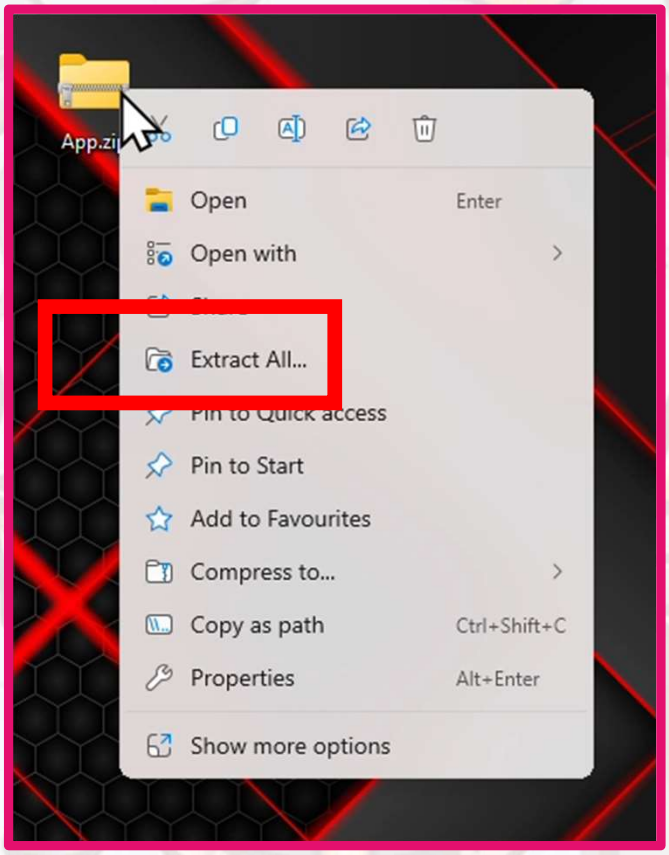


Let's get going!

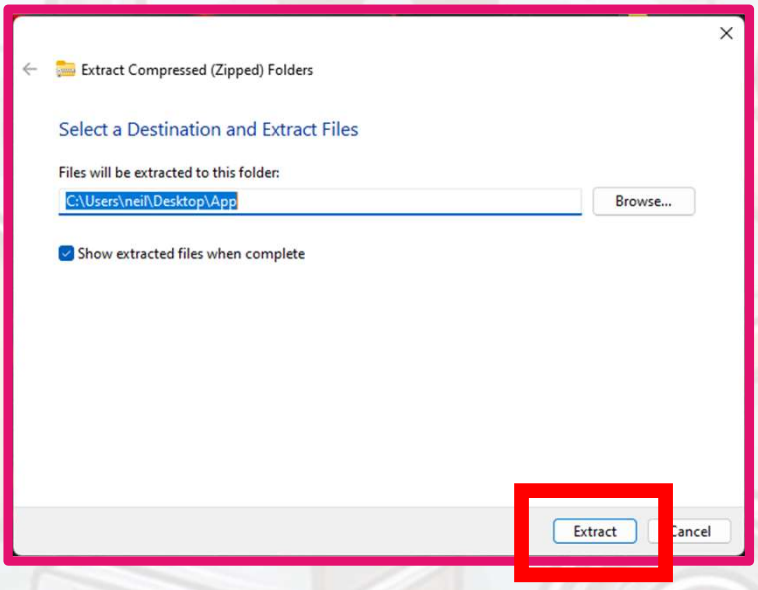
University of Sunderland

Extract the application!

Go to the desktop and find the file you just downloaded and right click on it.



Select **`Extract All`** and then **`Extract.`**

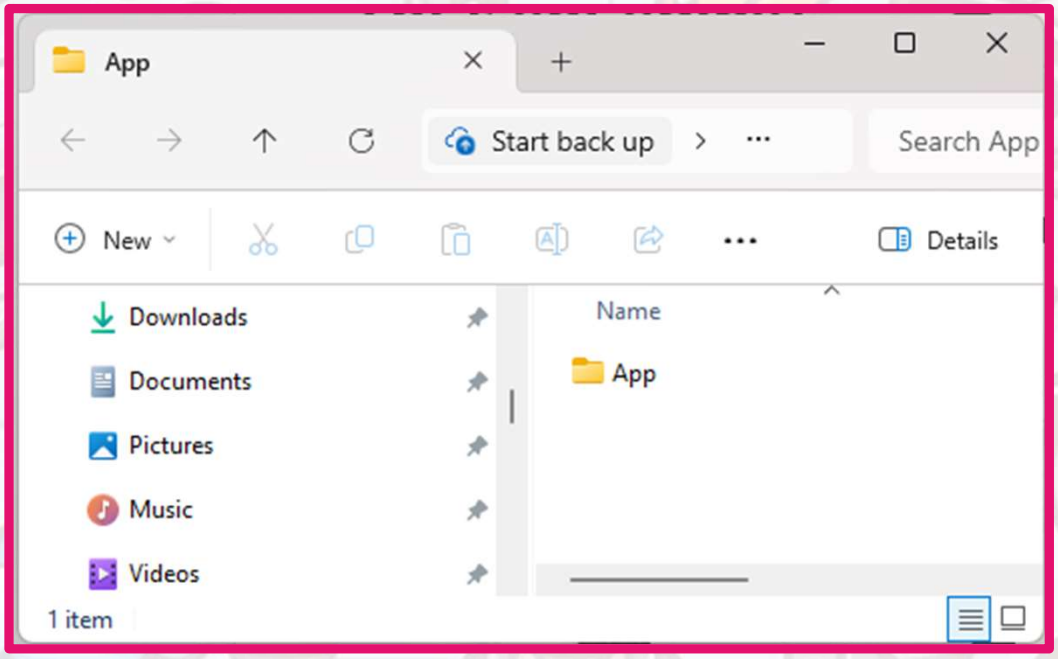


Let’s get going!

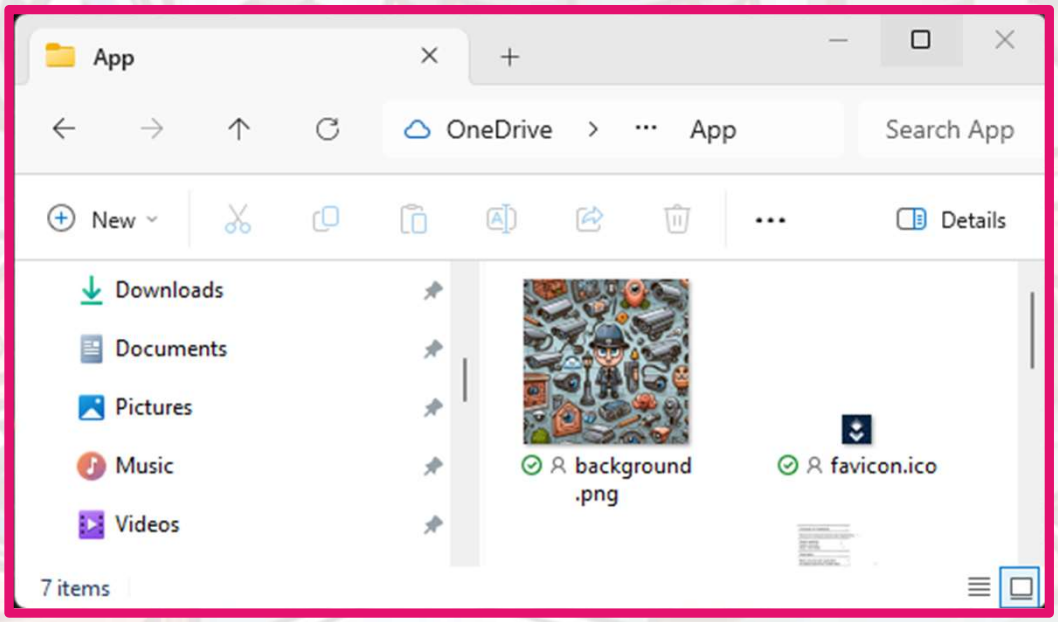
University of Sunderland

Running the application!

You can now see the Extracted application folder.



Go into the **App** folder by double clicking on it.

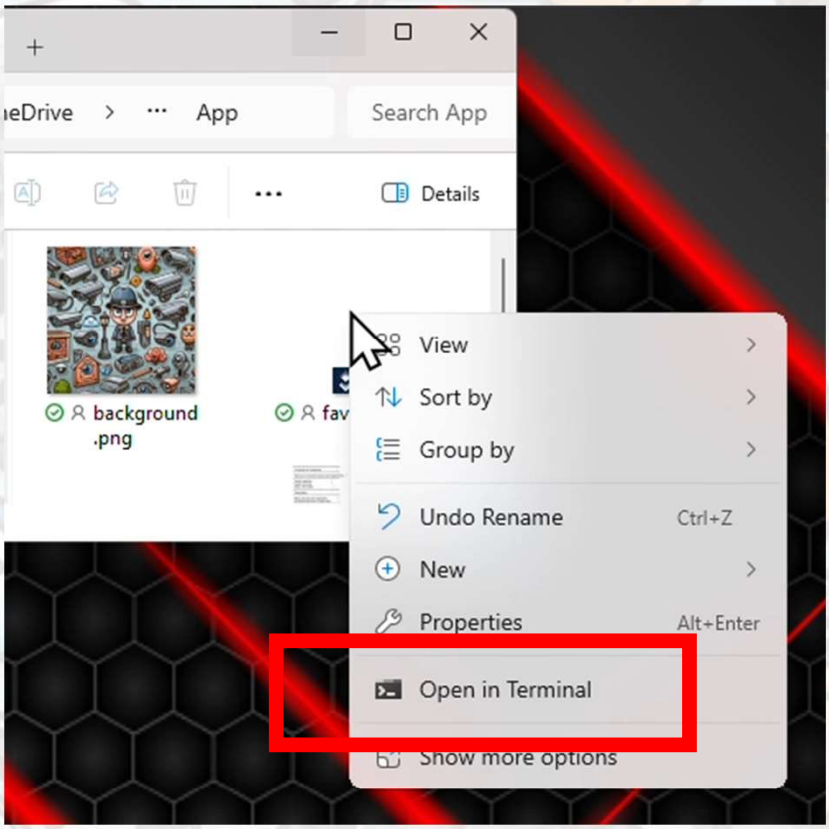


And now the Editor

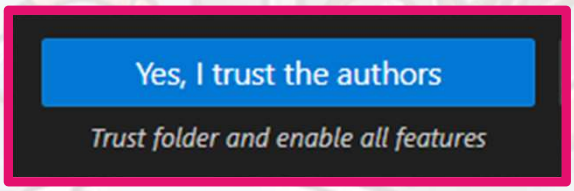
University of Sunderland

Running the application!

`Right Click` in the white space above the **`icon`** file to bring up the **`action menu`** as below:



Select the **`Open in Terminal`** option. When the terminal window opens go to it and type in "**`code .`**" (without the quotes, yes, a dot is required after **`code`**!). Press the **`RETURN`** key. You should now have **`VS Code`** running with the **`App`** folder showing the files. If you get a message asking if you Trust this folder select **`Yes.`**



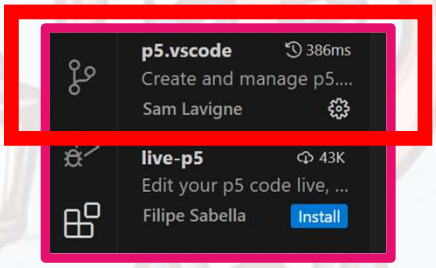
And now the Editor

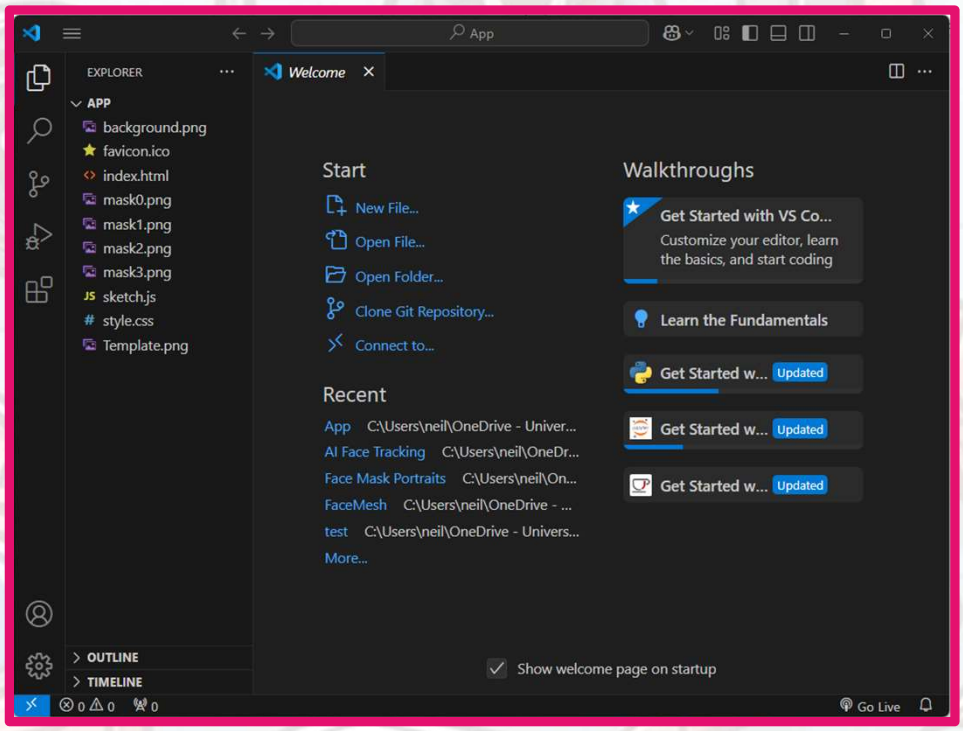
University of Sunderland

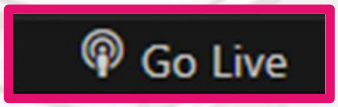
Running the application!

Now you can launch the application. At the bottom of the screen is an option "**`Go Live`**". This is the built in **`Live Server`** which is a local web server which runs the application.

If the "Go Live" option isn't displayed call the academic over, they may need to install the `p5.vscode` extension for you!





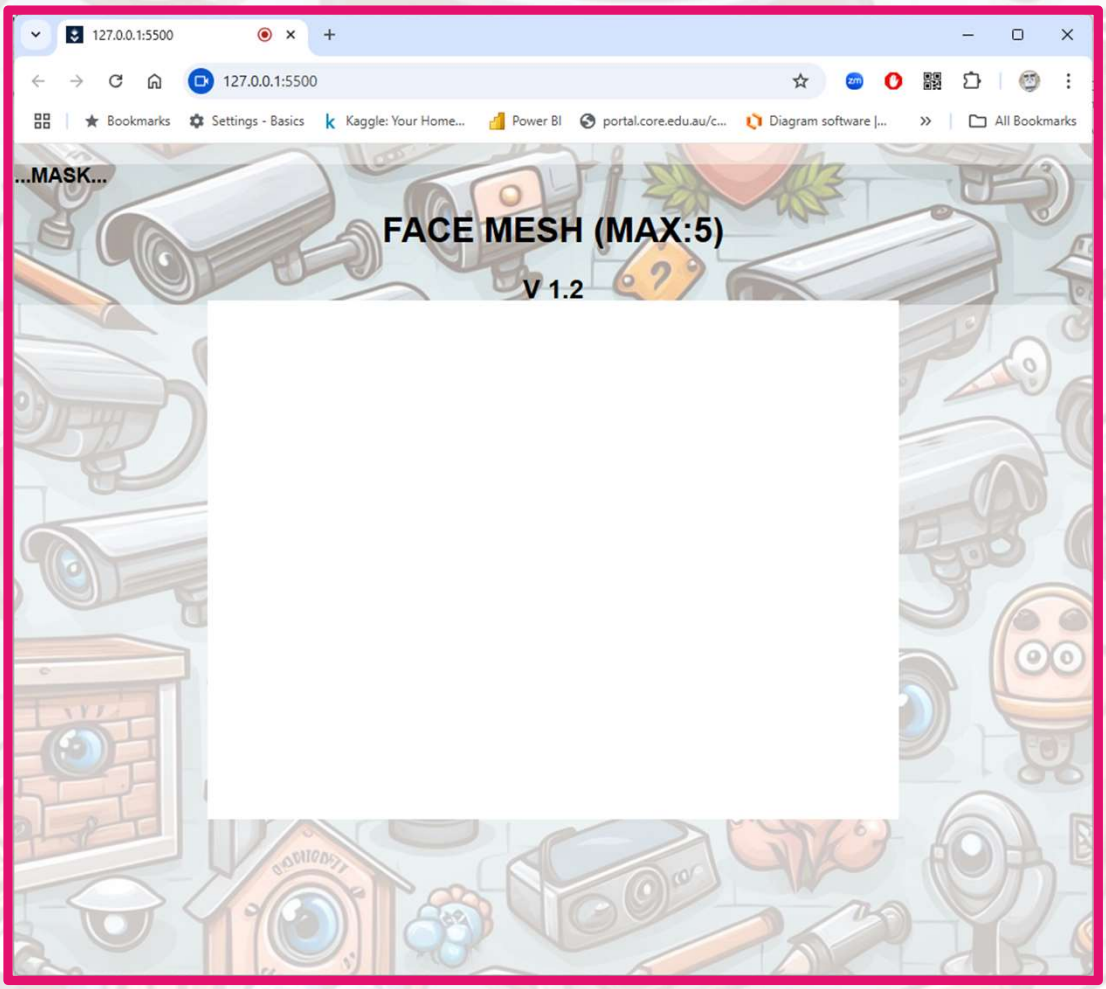


Development

University of Sunderland

Running the application!

This application will now be launched in the default browser and should look similar to the image below.



Development

University of Sunderland

Stage 1!

There is no functionality yet. Let's start by adding a menu, or at least some options, to get started!

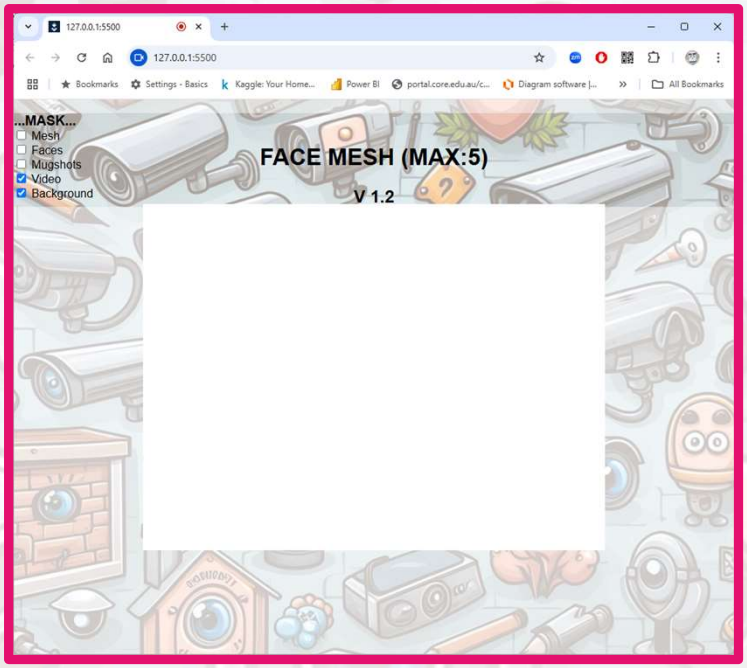
Copy the code below into the menu section of `function setup()` and save the code (`ctrl s`).

The application should automatically restart in the browser!
If it doesn't select the Go Live button again!

```
dotsCheckbox = createCheckbox(' Mesh');
dotsCheckbox.position(0, 40);
faceCheckbox = createCheckbox(' Faces');
faceCheckbox.position(0, 60);
mugshotsCheckbox = createCheckbox(' Mugshots');
mugshotsCheckbox.position(0, 80);
videoCheckbox = createCheckbox(' Video', true);
videoCheckbox.position(0, 100);
backCheckbox = createCheckbox(' Background', true);
backCheckbox.position(0, 120);
```

```
// 1 - Menu (Start)

// Menu (End)
```



Development

University of Sunderland

Stage 2!

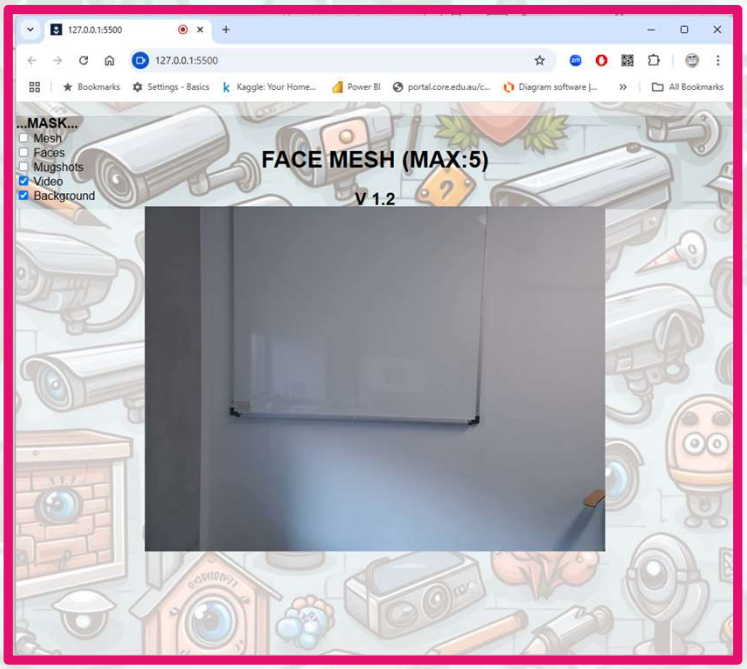
Next thing that needs to be done is to add a video feed to the application.

Copy the code below into the main body of the `function videoFeed()` save the code (`ctrl s`) and you should now be able to see a live feed from the webcam.

```
if (videoCheckbox.checked()) {
    image(video, 0, 0);
}
```

```
// 2 - Video (Start)

// Video (End)
```



If the browsers prompts for access to the camera select the *Allow* option.

Development

University of Sunderland

Stage 3!

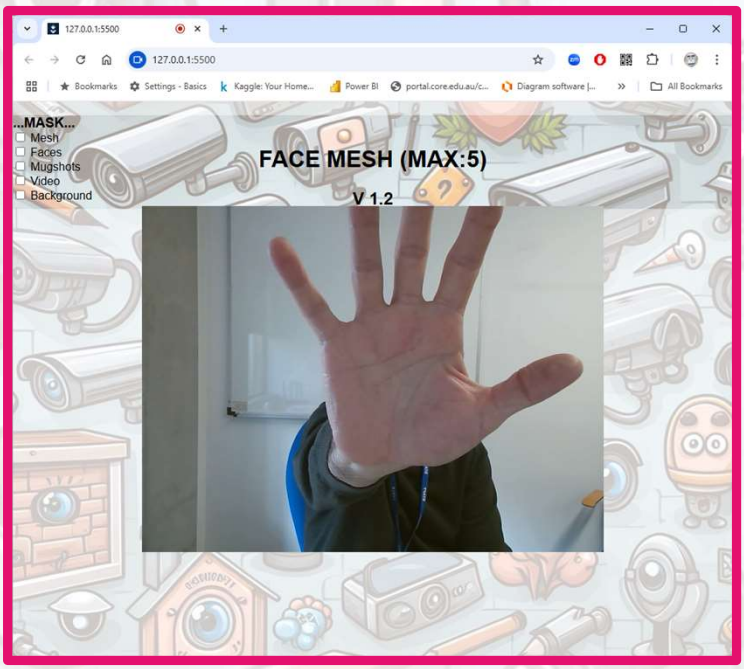
With the video feed working it will overwrite the entire background of the canvas, lets allow a user to disable the background. (Among other things, this will be handy for taking screen shots later!)

Copy the code below into the main body of `function back()` and save the code (`ctrl s`).

```
if (backCheckbox.checked()) {
   background(0);
}
```

```
// 3 - Background (Start)

// Background (End)
```



Have a play around with the `video` and `background` options. That looks like a freeze frame!

Development

University of Sunderland

Stage 4!

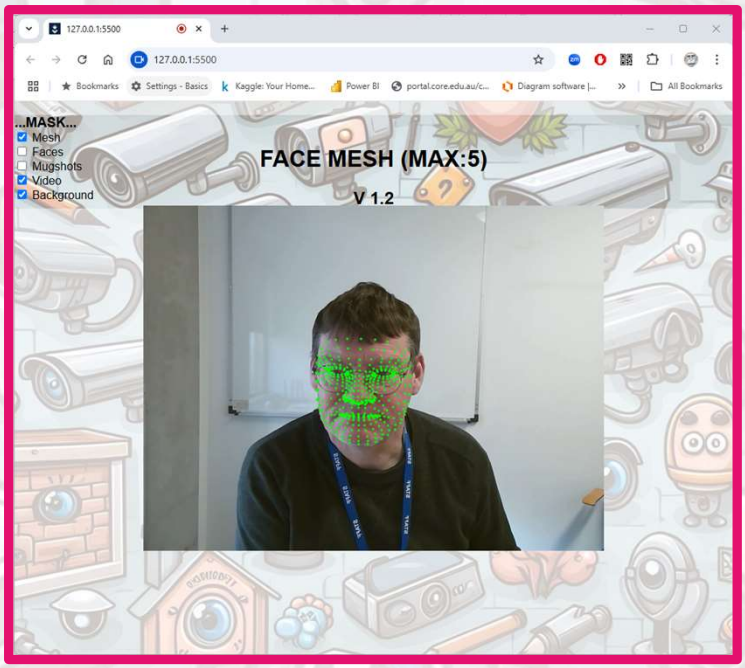
We can control the video and background so now let's add in the AI component and see what it can identify. The **`FaceMesh`** model will be picking up faces so let's show what it's found by highlighting the keypoints!

Copy the code below into the main body of `function dots()` and save the code (**`ctrl s`**).

```
if (dotsCheckbox.checked()) {
   if (faces.length > 0) {
     for (let loop1 = 0; loop1 < faces.length; loop1++) {
       let face = faces[loop1];
       for (let loop2 = 0; loop2 < face.keypoints.length; loop2++) {
         noStroke();
         fill(0, 255, 0);
         circle(face.keypoints[loop2].x, face.keypoints[loop2].y, 3);
       }
     }
   }
}
```

```
// 4 - Mesh Dots (Start)

// Mesh Dots (End)
```



Development

University of Sunderland

Stage 5!

Now we have the AI model detecting faces in the video feed we need to isolate them.

Copy the code below into the main body of `function isolateFaces()` and save the code (`ctrl s`).

```
if (faceCheckbox.checked()) {
    if (faces.length > 0) {
      let frame;
      let face;
      let box;
      let faceImage;
      for (let loop1 = 0; loop1 < faces.length; loop1++) {
        frame = video.get(0, 0, width, height);
        face = faces[loop1];
        box = face.box;
        shape = createGraphics(width, height);
        shape.fill(0);
        shape.strokeWeight(5);
        shape.beginShape();
        for (let points = 0; points <
          face.faceOval.keypoints.length; points++) {
          shape.point(face.faceOval.keypoints[points].x,
          face.faceOval.keypoints[points].y);
          shape.curveVertex(face.faceOval.keypoints[points].x,
          face.faceOval.keypoints[points].y);
        }
        shape.endShape(CLOSE);
        frame.mask(shape);
        let faceImage = frame.get(box.xMin, box.yMin, box.width, box.height);
        image(faceImage, box.xMin, box.yMin);
      }
    }
  }
```

```
// 5 - Isolate Faces (Start)

// Isolate Faces (End)
```

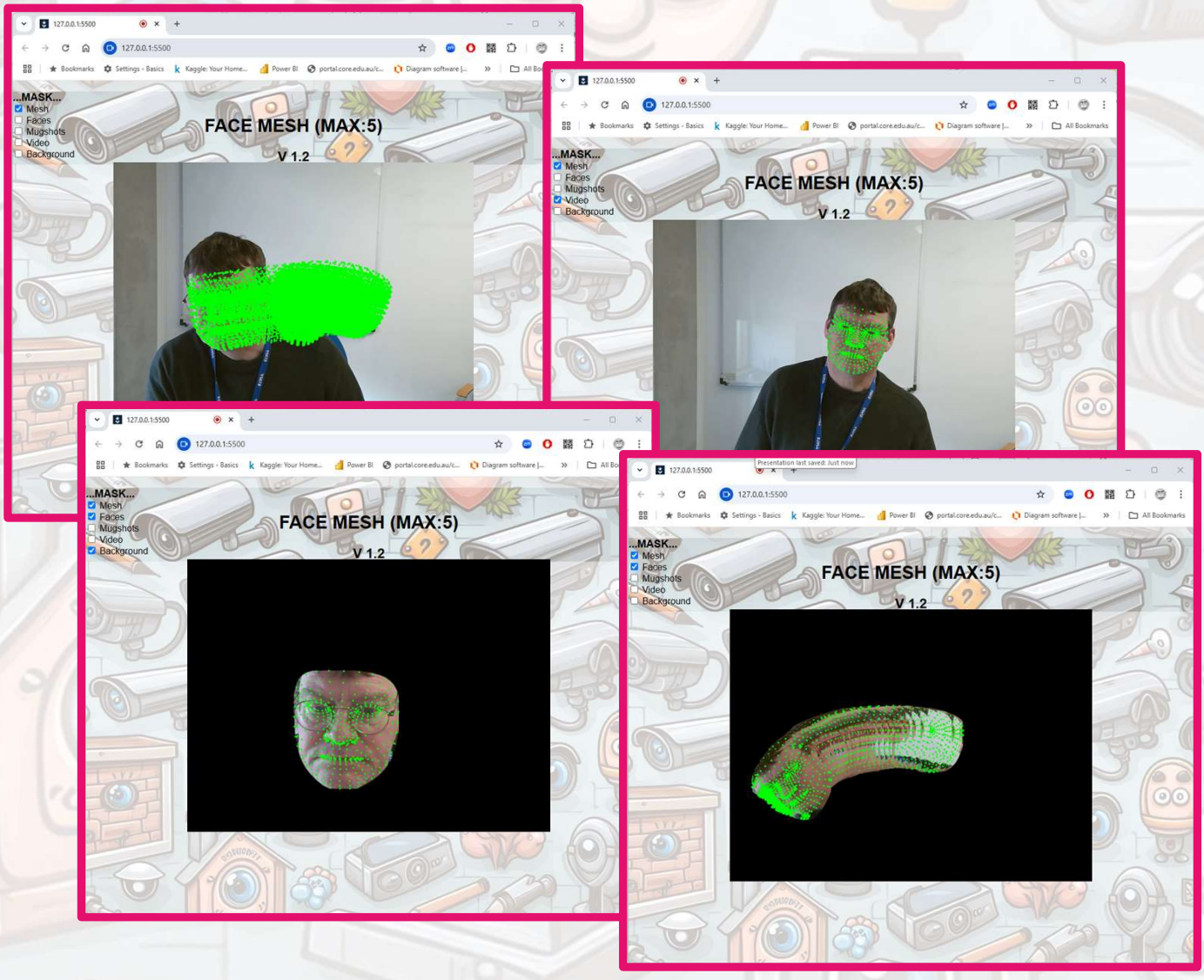
Development

University of Sunderland

Stage 5!

You have quite a lot of functionality now so try out the various combinations and see what you can do so far.

There are some strange effects starting to emerge now but not quite what we need for a serious security tool, but you never know when something might emerge. Two more features may be helpful. What about getting some mugshots and saving images (faces and video snapshots).



Development

University of Sunderland

Stage 6!

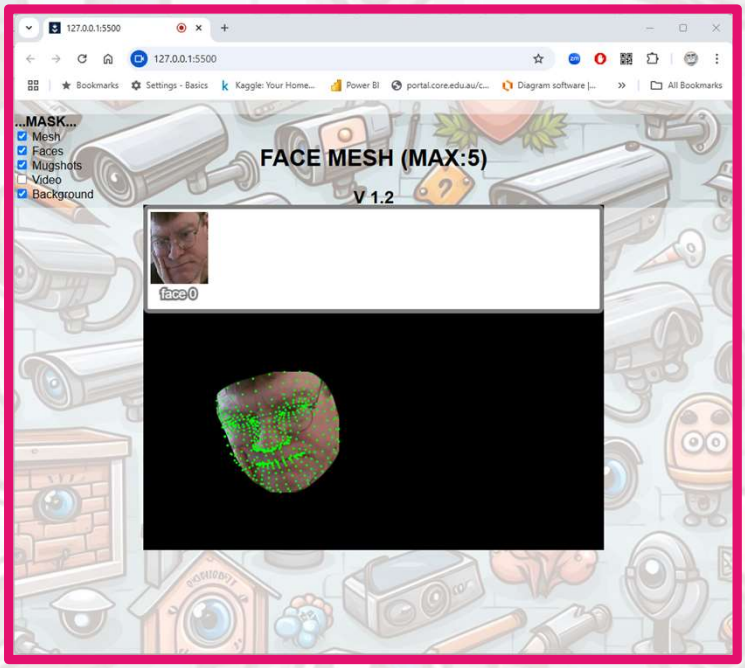
Having isolated faces lets get them laid out in a rouge's gallery across the top of the screen!

Copy the code below into the main body of `function mugs()` and save the code (`ctrl s`).

```
if (mugshotsCheckbox.checked()) {
  strokeWeight(6);
  stroke(125, 125, 125);
  fill(255)
  rect(3, 3, 634, 145, 5);
  for (let loop1 = 0; loop1 < faces.length; loop1++) {
    let face = faces[loop1];
    let box = face.box;
    let faceImage = video.get(box.xMin, box.yMin, box.width, box.height);
    faceImage.resize(80, 100);
    textSize(18);
    image(faceImage, (loop1 * 85) + 10, 10);
    text("face " + loop1, (loop1 * 85) + 25, 130);
  }
}
```

```
// 7 Save (Start)

// Save (End)
```



Development

University of Sunderland

Stage 7!

Finally let's implement saving the rouge's gallery (press **g**) and taking a snapshot of the screen (press **s**)

Copy the code below into the main body of the **sketch.js** file and save the code (**ctrl s**).

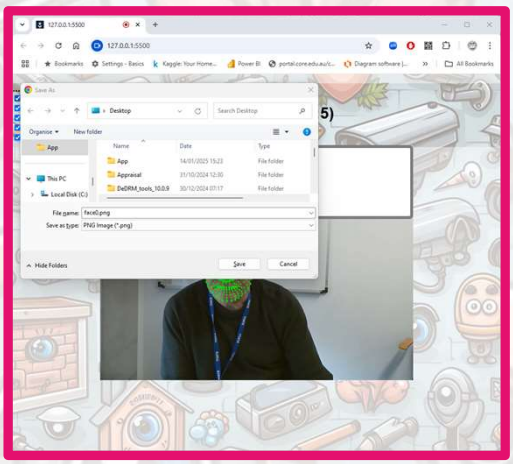
```
function saveFaces() {
  if (mugshotsCheckbox.checked()) {
    for (let loop1 = 0; loop1 < faces.length; loop1++) {
      let face = faces[loop1];
      let box = face.box;
      faceImage = video.get(box.xMin, box.yMin, box.width, box.height);
      faceImage.save('face' + loop1, 'png');
    }
  }
}

function saveSnapshot() {
  saveCanvas("canvas.png");
}

function keyPressed() {
  if (key === 'm') {
    saveFaces();
  }
  if (key === 's') {
    saveSnapshot();
  }
}
```

```
  // 7 Save (Start)

  // Save (End)
```



Deployment!

University of Sunderland

Let’s use your new AI powered application!

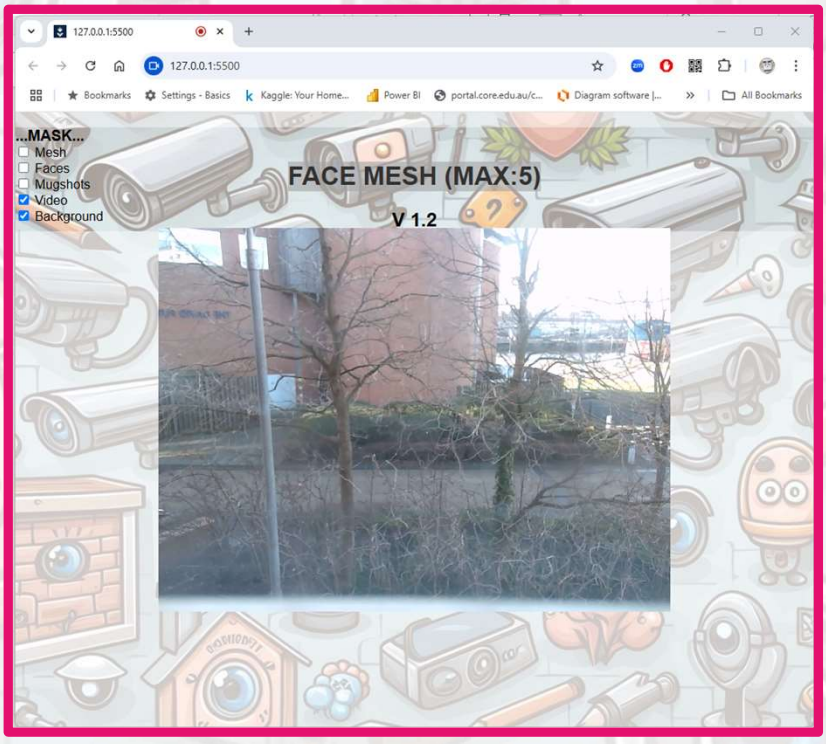
Now the application is complete (for now) with all the functionality it’s time to take it for a test ride to see what it can do.

You may have noticed that the title on the page is:

FACE MESH (MAX:5)

That’s because it is set to identify a maximum of 5 faces in the video feed. So how about getting more faces in front of the camera.

What about not using the webcam pointing directly at you and point it somewhere else. Now the application is a security camera console!



And Finally!

University of Sunderland

Creating basic applications with AI is just the tip of the iceberg.

AI now empowers us to revolutionize industries, enhance everyday life, and solve complex problems in ways we never thought possible.

What we have seen here is just a simple security application being built from the ground up.

Future possibilities are vast and exciting, and you can be a part of it!

University of Sunderland

On the **BSc Computer Science** Programme at the **University of Sunderland** you'll study AI and Machine Learning in more detail, which means you'll be able to create some amazing things and have a good grasp on this important emerging technology.

BSc Computer Science

Other Programmes that may be of interested

BSc Cybersecurity and Digital Forensics

BSc Games Development