

1) Connecting to drive

```
from google.colab import drive
drive.mount('/content/drive')

Mounted at /content/drive
```

2) Building dataset:

```
from IPython.display import display, Javascript
from google.colab.output import eval_js
from base64 import b64decode
import uuid

def take_photo(quality=0.8):
    js = Javascript('''
        async function takePhoto(quality) {
            var finish = false;
            const div = document.createElement('div');
            var input_name = document.createElement("input");
            input_name.placeholder = "Enter a position"
            const capture = document.createElement('button');
            capture.textContent = 'Capture';
            const stop = document.createElement('button');
            stop.textContent = 'Stop';
            stop.onclick = function() {finish = true; capture.click()};
            div.appendChild(input_name)
            div.appendChild(capture);
            div.appendChild(stop);

            const video = document.createElement('video');
            video.style.display = 'block';
            const stream = await navigator.mediaDevices.getUserMedia({video: true});

            document.body.appendChild(div);
            div.appendChild(video);
            video.srcObject = stream;
            await video.play();

            // Resize the output to fit the video element.
            google.colab.output.setIframeHeight(document.documentElement.scrollHeight, tr

            // Wait for Capture to be clicked.
            await new Promise((resolve) => capture.onclick = resolve);
            if(finish){
                stream.getTracks().forEach(function(track) {
                    track.stop();
                });
                div.innerHTML = ""
                return null
            }
        }
    ''')
```

```

        const canvas = document.createElement('canvas');
        canvas.width = video.videoWidth;
        canvas.height = video.videoHeight;
        canvas.getContext('2d').drawImage(video, 0, 0);
        stream.getVideoTracks()[0].stop();
        div.remove();
        return {data: canvas.toDataURL('image/jpeg', quality), name: input_name.value
    }
    '')
display(js)
photo = eval_js('takePhoto({})'.format(quality))
if photo is None:
    return None, None
data = photo['data']
name = photo['name']
binary = b64decode(data.split(',')[1])
filename = str(uuid.uuid1())+".jpg"
with open(filename, 'wb') as f:
    f.write(binary)
return filename, name

from IPython.display import Image

try:
    while True:
        filename, name = take_photo()
        if filename is None:
            break
        print('Saved to {}'.format(filename))
        if name == "":
            folder = "other"
        else: folder = name
        !mkdir -p drive/Shareddrives/Projet-ML-M2-PLS/data/positions/train/$folder && m

    # Show the image which was just taken.
    #display(Image(filename))
    print('Camera stopped')
except Exception as err:
    # Errors will be thrown if the user does not have a webcam or if they do not
    # grant the page permission to access it.
    print(str(err))

```

Saved to 6cfbb408-710e-11eb-8522-0242ac1c0002.jpg
Saved to 705c65a2-710e-11eb-8522-0242ac1c0002.jpg
Saved to 736b1afe-710e-11eb-8522-0242ac1c0002.jpg
Saved to 75feaf4c-710e-11eb-8522-0242ac1c0002.jpg
Saved to 7a2b1dda-710e-11eb-8522-0242ac1c0002.jpg
Saved to 7d600f9c-710e-11eb-8522-0242ac1c0002.jpg
Saved to 873c9134-710e-11eb-8522-0242ac1c0002.jpg
Saved to 8a305a4c-710e-11eb-8522-0242ac1c0002.jpg
Saved to 8de4b5ac-710e-11eb-8522-0242ac1c0002.jpg
Saved to 90a57d12-710e-11eb-8522-0242ac1c0002.jpg
Saved to 933805c2-710e-11eb-8522-0242ac1c0002.jpg
Saved to 96793cba-710e-11eb-8522-0242ac1c0002.jpg
Saved to 9e735bee-710e-11eb-8522-0242ac1c0002.jpg
Saved to aa373680-710e-11eb-8522-0242ac1c0002.jpg
Saved to adb548c4-710e-11eb-8522-0242ac1c0002.jpg
Saved to b0bc9c16-710e-11eb-8522-0242ac1c0002.jpg
Saved to b38f4358-710e-11eb-8522-0242ac1c0002.jpg
Saved to b5dd3f84-710e-11eb-8522-0242ac1c0002.jpg
Saved to b95143b8-710e-11eb-8522-0242ac1c0002.jpg
Saved to bd19065c-710e-11eb-8522-0242ac1c0002.jpg
Saved to c7dlb90e-710e-11eb-8522-0242ac1c0002.jpg
Saved to ce7be0f4-710e-11eb-8522-0242ac1c0002.jpg
Saved to d4f35520-710e-11eb-8522-0242ac1c0002.jpg
Saved to d8d6674a-710e-11eb-8522-0242ac1c0002.jpg
Saved to e5a4b6a2-710e-11eb-8522-0242ac1c0002.jpg
Saved to e8dd83b2-710e-11eb-8522-0242ac1c0002.jpg
Saved to ec3ec2be-710e-11eb-8522-0242ac1c0002.jpg
Saved to f1abf050-710e-11eb-8522-0242ac1c0002.jpg
Saved to f57f8e4e-710e-11eb-8522-0242ac1c0002.jpg
Saved to f976c468-710e-11eb-8522-0242ac1c0002.jpg
Saved to fe4d31a2-710e-11eb-8522-0242ac1c0002.jpg
Saved to 01bc8824-710f-11eb-8522-0242ac1c0002.jpg
Saved to 0898dc42-710f-11eb-8522-0242ac1c0002.jpg
Saved to 15fe80bc-710f-11eb-8522-0242ac1c0002.jpg
Saved to 1a6c0c32-710f-11eb-8522-0242ac1c0002.jpg
Saved to 218a6d06-710f-11eb-8522-0242ac1c0002.jpg
Saved to 68ba586c-710f-11eb-8522-0242ac1c0002.jpg
Saved to 6ca2a452-710f-11eb-8522-0242ac1c0002.jpg
Saved to 7006666a-710f-11eb-8522-0242ac1c0002.jpg
Saved to 775d24ee-710f-11eb-8522-0242ac1c0002.jpg
Saved to 7b280eae-710f-11eb-8522-0242ac1c0002.jpg
Saved to 9e1ab308-710f-11eb-8522-0242ac1c0002.jpg
Saved to a0923f98-710f-11eb-8522-0242ac1c0002.jpg
Saved to a41d842e-710f-11eb-8522-0242ac1c0002.jpg
Saved to ce6de174-710f-11eb-8522-0242ac1c0002.jpg
Saved to d227569c-710f-11eb-8522-0242ac1c0002.jpg
Saved to d5430056-710f-11eb-8522-0242ac1c0002.jpg
Saved to d7a0a0b0-710f-11eb-8522-0242ac1c0002.jpg
Saved to da981f32-710f-11eb-8522-0242ac1c0002.jpg
Saved to dd5e4d54-710f-11eb-8522-0242ac1c0002.jpg
Saved to f2a590c8-710f-11eb-8522-0242ac1c0002.jpg
Saved to f57a85ce-710f-11eb-8522-0242ac1c0002.jpg
Saved to f8afaecc-710f-11eb-8522-0242ac1c0002.jpg
Saved to fcb5aff8-710f-11eb-8522-0242ac1c0002.jpg

