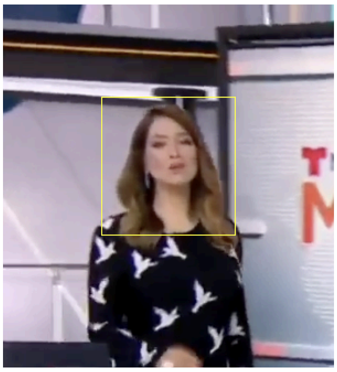
Blur That Guy

Goal

Journalists often need to alter footage, for instance, to preserve the anonymity or location of interviewees or to hide a logo, license plate, etc. The objective of this programming task is to build a system that someone could use to take a video file, select a face, and display the video with the selected face hidden.

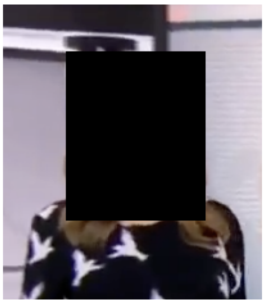
The recommended approach would be something like this:

1) Extract the video frames as images:
2) Run some open source computer vision tool to detect where faces are in each image:



3) Try to correlate the faces from one image to the next, to be able to follow a face when a person moves or the camera zooms in or out:

4) Provide a UI that allows the end user to view the video with the face bounds, select a face, and hide it either by blurring or covering with black:



Minimal requirements

The system can process any .mp4 video file that can be played in a web browser.

The system must allow tracking one face within one shot. For instance, if the footage shows person A, then person B, then person A again, selecting person A on its first appearance should track it up until person B appears. If you manage to track a person across shot changes, that is ok, but this is not part of the requirement.

The UI you create to view the original video, select a face, and view the resulting video with the face hidden must be a web UI that works at least in Chrome.

All the tools used must be open source. This includes the tools used to process the video and the images, as well as the tools needed to build and run your project.

The system must run standalone on a computer without network access, i.e., it must not rely on external services or resources.

Expectations

How you decide to build a system that solves the problem is entirely up to you.

You could, for instance, make a web server exposing a web page where a user could upload a video, process it, and then click on faces directly in the video to select them.

You could also make a command-line tool that takes the video as an argument, processes it, and writes the web page to an .html file to open for preview and to select faces.

For the actual hiding of faces, you could do this by placing canvases on top of a regular video player in the UI, you could render a second video file where the frames would be modified to hide the faces, you could decode the video using web codec APIs and update the canvas in memory for each frame before displaying it, etc.

As long as the minimal requirements are met, you are free to do whatever you want, using whatever technologies you want.

Delivery

We expect you to deliver your work as a github repository. It can be a public one, but if you prefer to use a private one, that’s fine too. In that case, you will need to invite the “sebmjoll” github user.

The repository description should be sufficient to know how to build, run and use your system.

Your system may be tested on Windows, Linux, or MacOS, either on an Intel CPU or on a Silicon one. As a consequence, you need to be careful about your requirements. For instance, having a file named “mystuff.py” and instructions just saying “Run mystuff.py” is very likely to fail unless the tester has exactly the same python setup as you do. A good way to make sure your system can work on any machine could be to design it as a Docker Linux container and include the Dockerfile in your repository so that we could build and run your Docker image.

Also, we will have a look at how your commits are organized, so you may want to keep a relatively clean history without commits like “Did stuff”, “Fixed broken stuff”, “Fixed broken stuff 2”, etc.

Find your superpower and show it off

Depending on what you like to do and what you are good at, feel free to dig deeper in any aspects of the project.

For instance, if you prefer frontend, you could focus on delivering a beautiful UI with great user experience and show your skills with a frontend framework like Vue or React.

If you’re interested in AI tools, you could focus on making the face tracking as good as possible, for example across scene changes, or you could focus on making the face tracking as fast as possible.

If you’re interested in video processing, you may want to look at supporting different video formats.

If you fancy infrastructure work, you may want to build your system in such a way that it could also be deployed on a cloud platform like AWS or GCP.

If you like hardcore challenges, you could try making something that also works on a live video stream.

Etc, etc.
Go as wild as you want. And have fun building it.