Software design and development year 12

Assessment task 1

Noah Hallows

Identification of the problem:

In the modern world there are a significant number of photos taken by people all the time. This poses significant privacy and possible safety concerns as anyone will be able to access information about your appearance, location history and friends or someone could use these pictures to make sexually explicit content of you using AI. However, searching through large numbers of photos to remove your face is not a viable solution. My solution is an automated system that uses open-source face recognition to search through large numbers of images and then deletes or edits the images with the users face to anonymize them.

Generation of ideas:

I have always been concerned with my digital privacy and since I have stopped being home schooled, I have had significantly more photos of me taken by other people. I found inspiration for my system when I was watching a Japanese cyberpunk anime one of the characters would hack computer systems to place a logo over their face to preserve their anonymity, which gave me the idea to do a similar thing with python.

Requirements of the solution

- Accurately identify faces of any demographic
- Accurately blur/remove these faces without destroying the whole image
- Only run on the specified images
- Have a low bar to use
- Run quickly on small numbers of images

Communication with others:

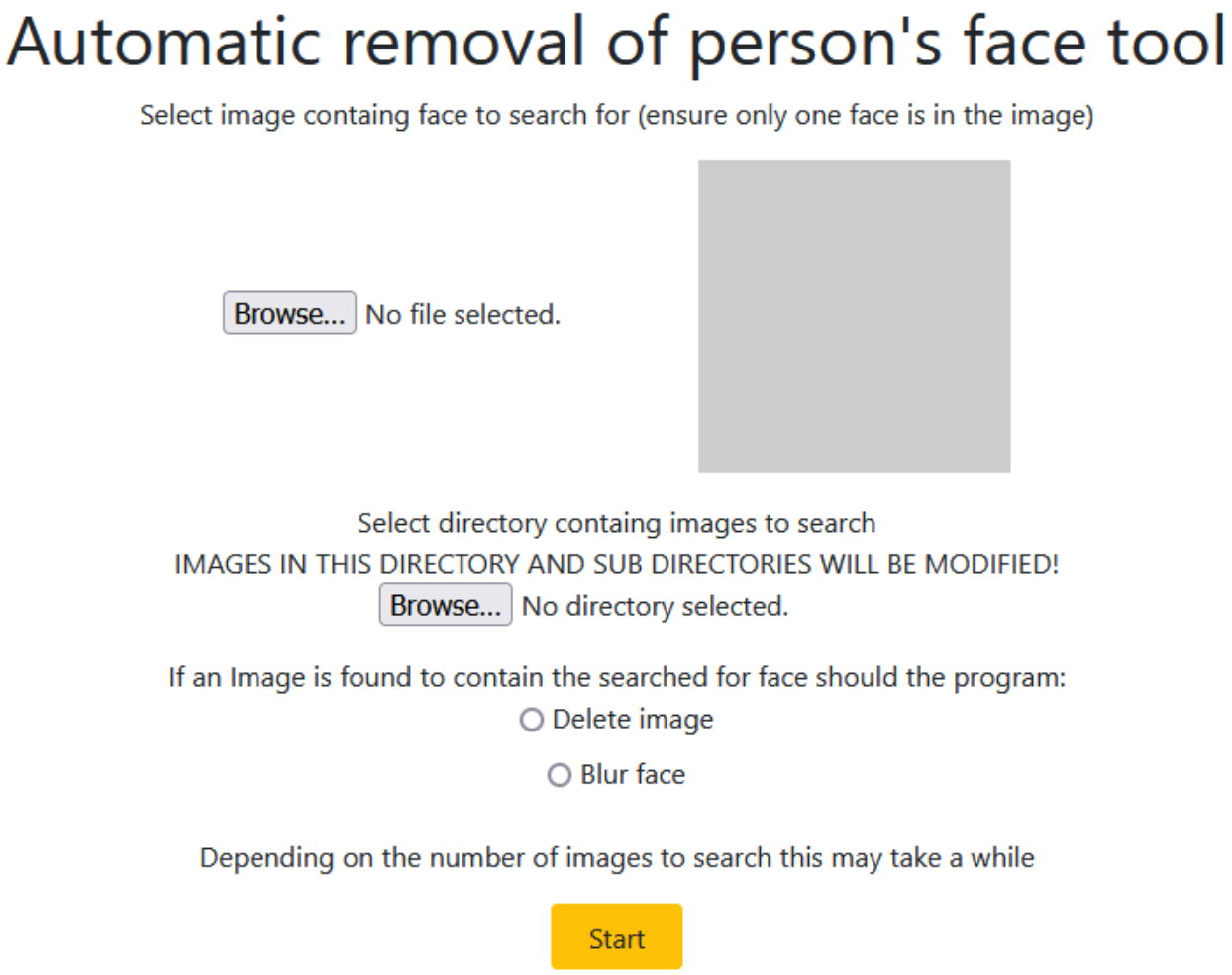
- Ana Williams (teacher)
- Brendon Godfrey
- Zeek Mattaronno
- John Hallows (father)

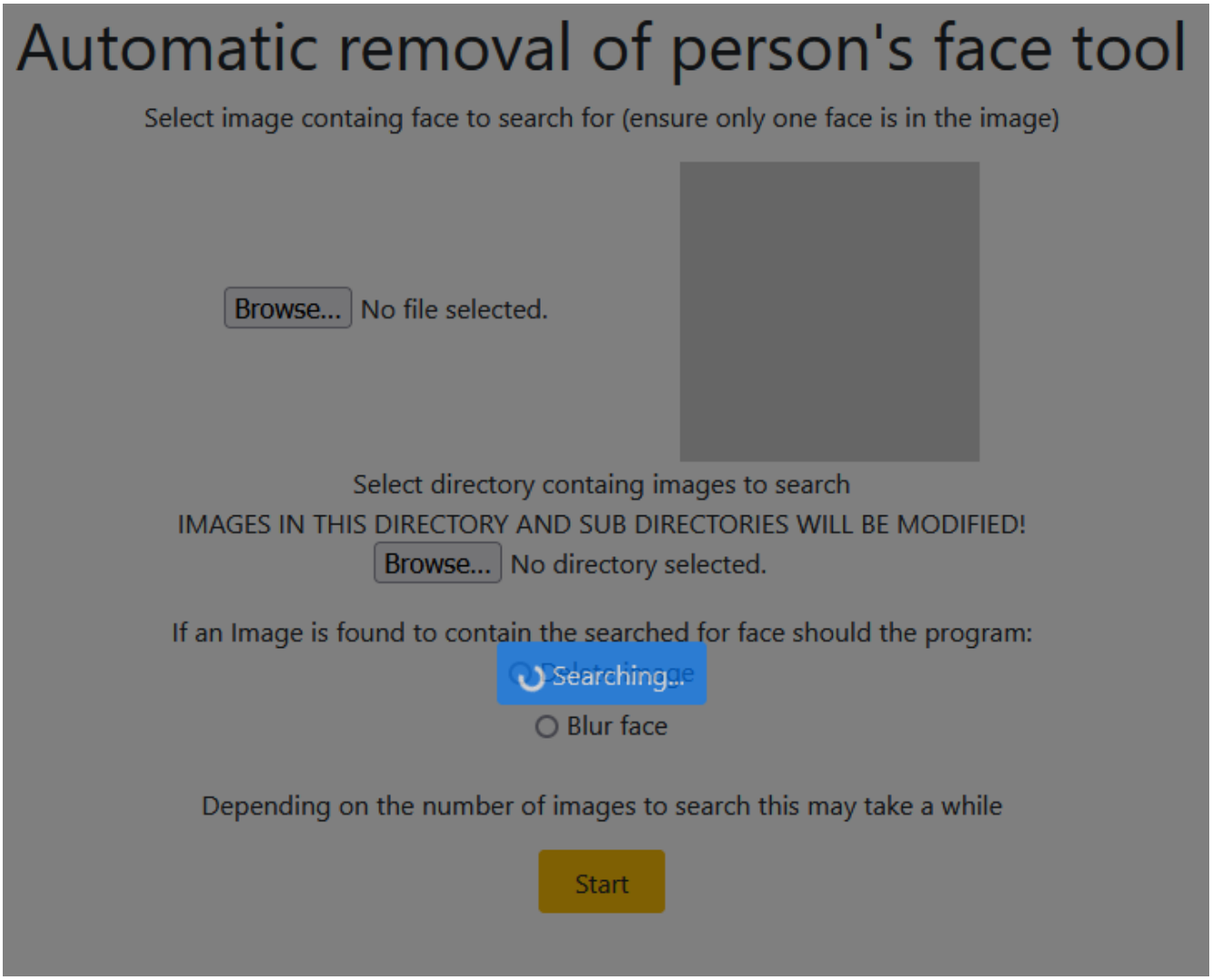
Selection of appropriate data types:

This software solution will use two main data types, strings to store the location of images and numpy arrays to store the processed images.

Draft interface design:

I used html, css and javascript to make a mockup interface design the screenshot doesn’t display the whole design/





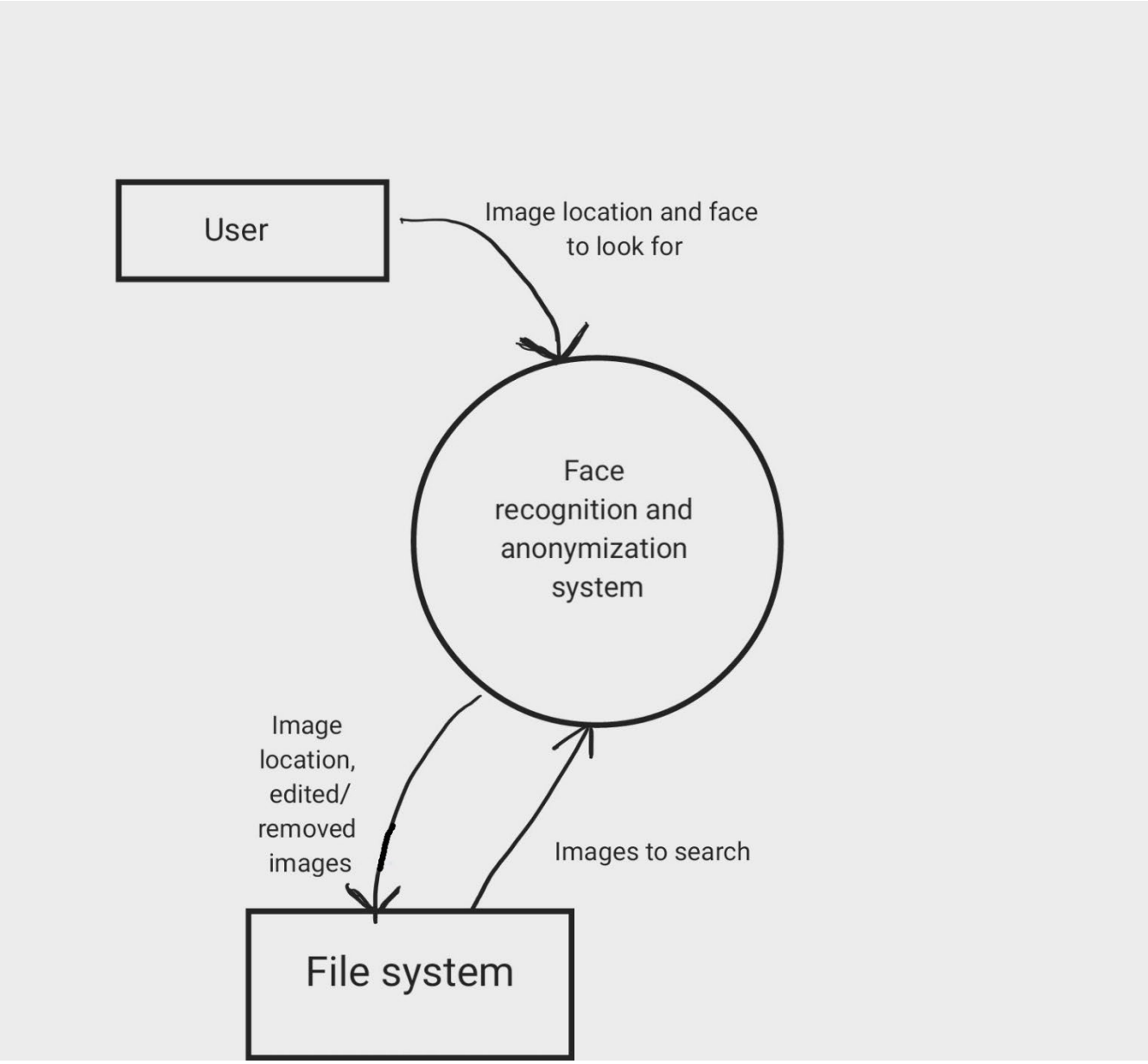
Social and Ethical issues:

Face recognition is the topic of significant debate about its potential for misuse. As with most projects containing face recognition my project has the potential for abuse however I have taken steps to minimize this through only identifying the face of the individual searched for, meaning this can't be used for surveillance, and using open-source software to ensure non authorized people aren’t using my project for nefarious purposes. Additionally, there have been issues with face recognition software not recognizing people who are of different demographics to the developers, I will run extensive testing to reduce these issues as much as possible in my software. This program could accidentally remove the wrong person from pictures which could cause harm, this could be mitigated by adding a manual confirmation system to the program.

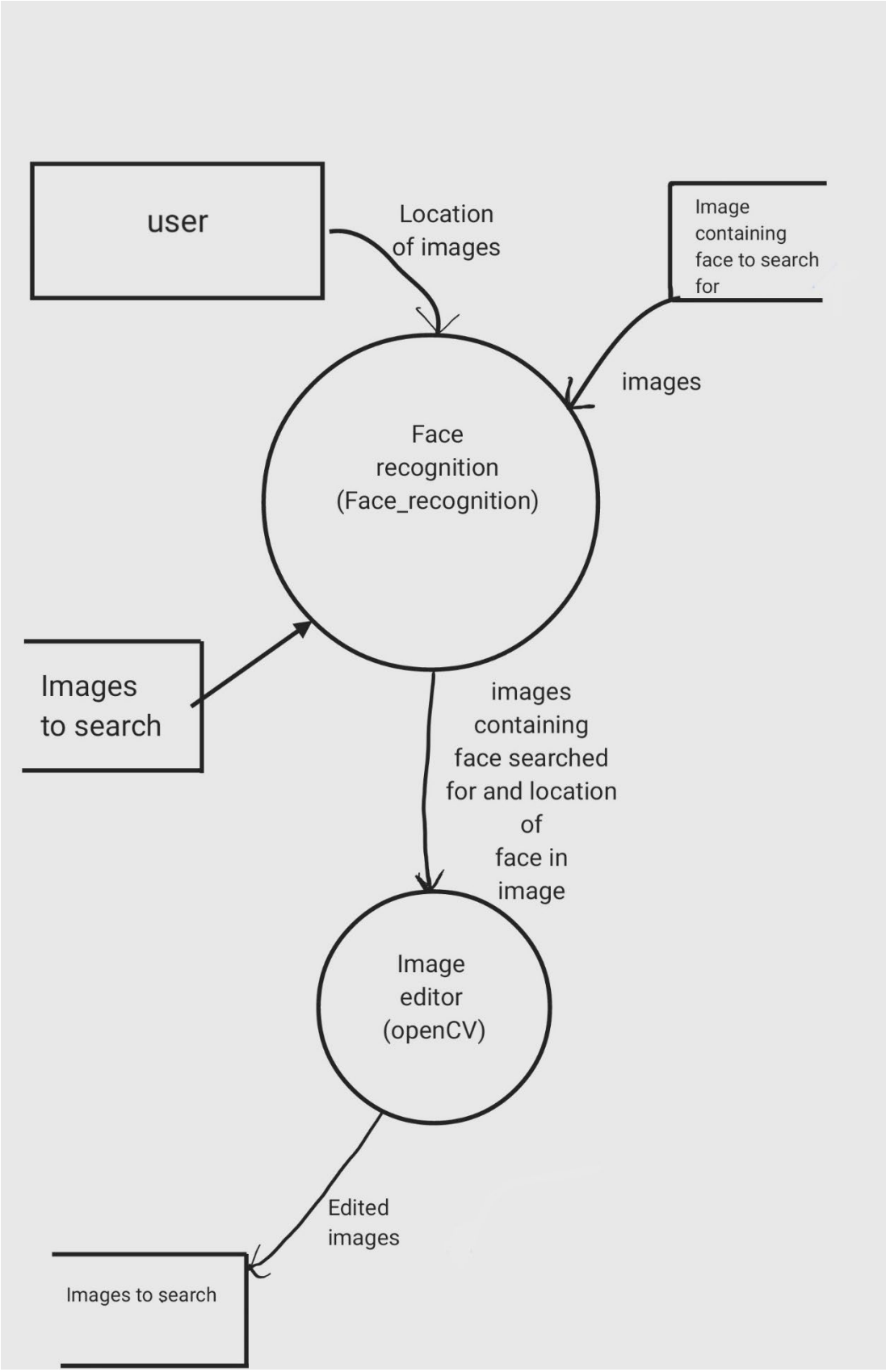
IPO diagram

Input	Processing	Output
Image of person to find	Run face recognition on image	
Blur or delete images containing face searched for		
Location of images to search	If image contains the person the program is looking for edit the image according to the previous input	Edited images

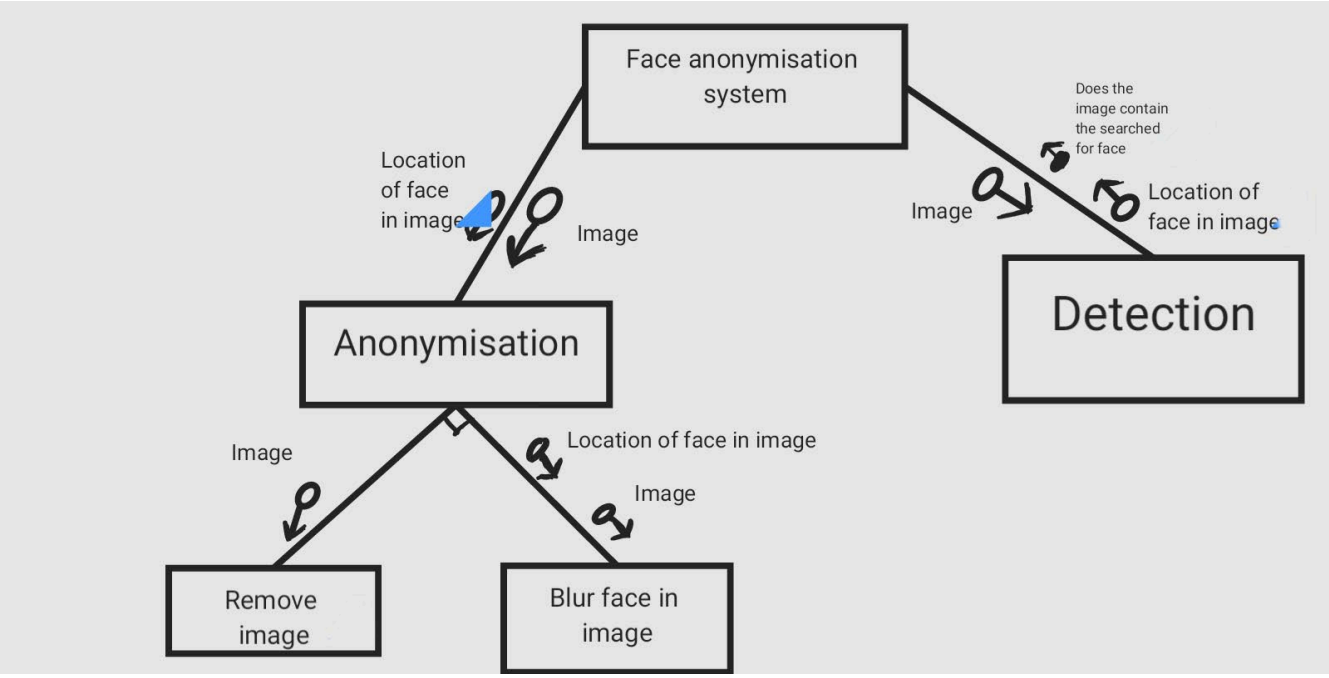
Context diagram



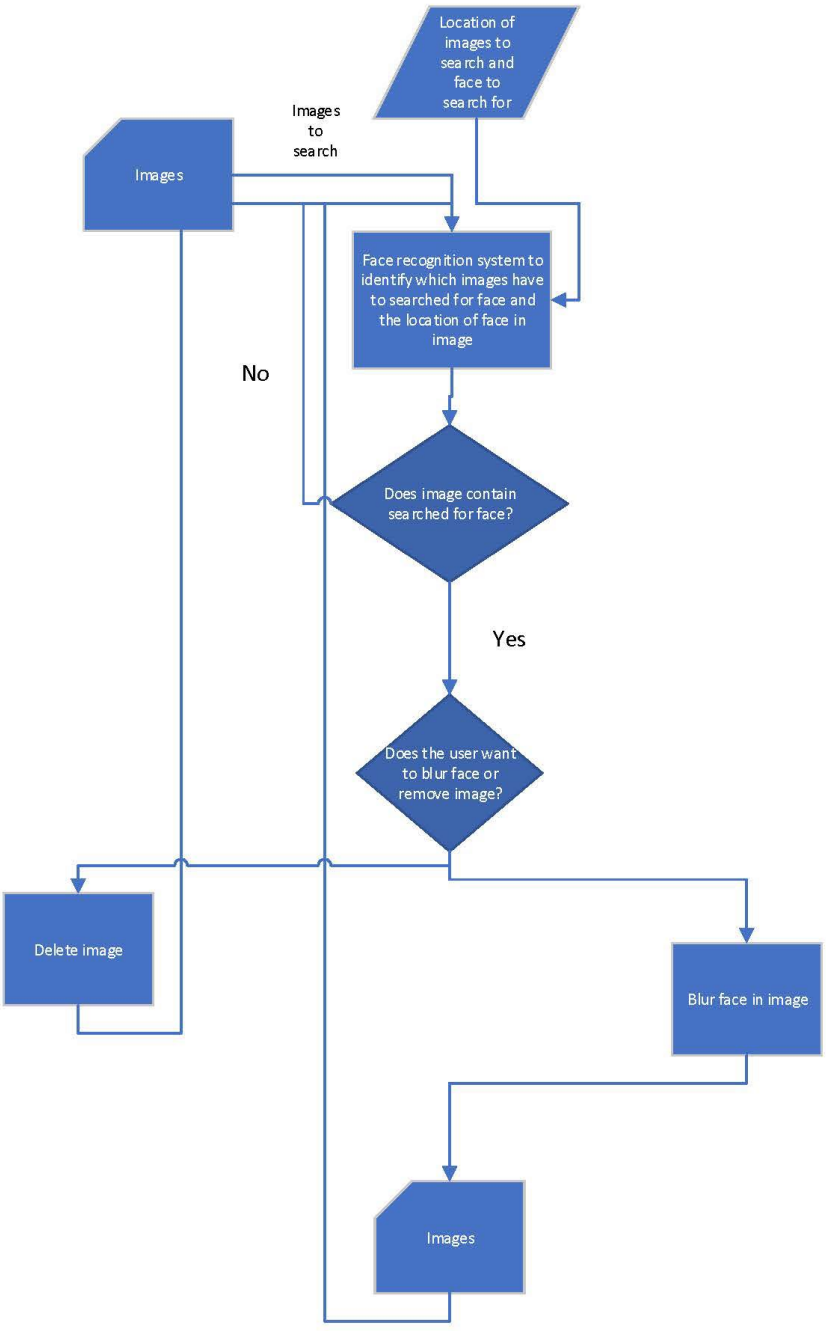
Data flow diagram



Structure chart



System flow chart



Data Dictionary

item	Data type	Description	validation
face_to_search_for	string	String that stores the location of the image containing the face to search for	File exists, valid file directory format
images_to_search	string	String that stores the directory of images to search	Directory exists, valid file directory format
face_to_search_for	Image(eg .png)	Image containing face to search for	Checks the face contains an image, checks it's a valid image file
Image_to_search	Image(eg .png)	Image to search	Check it's a valid image file
edited_image	Image(eg .png)	Edited version of "image_to_search" to remove face searched for	Check it's a valid image file
blur	boolean	True/false value to specify if the program should delete the image or blur the face	NA

Planning and designing

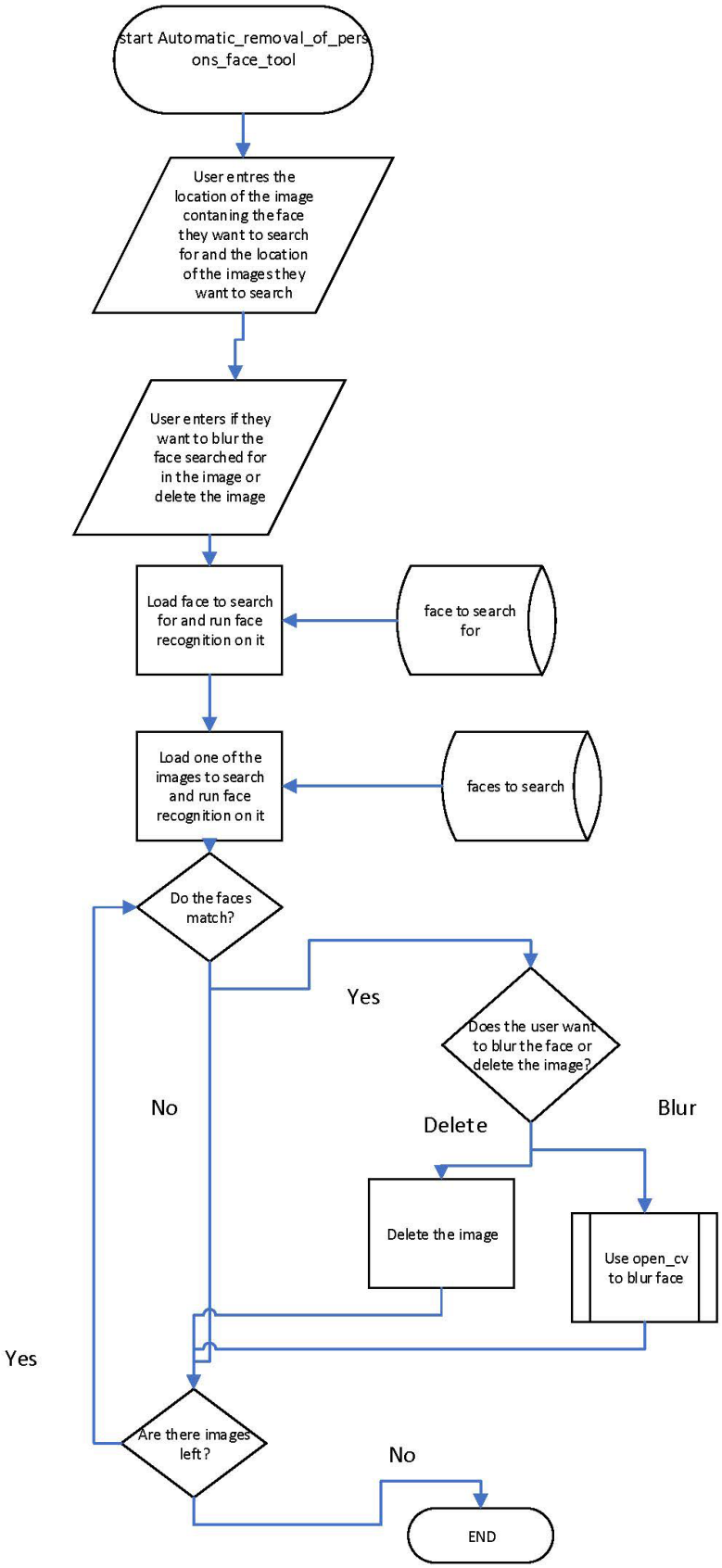
Algorithm

Pseudocode

```
BEGIN Automatic_removal_of_persons_face_tool

DISPLAY("Enter the location of the image containg the face to search for: ")
face_to_search_for_location <- INPUT()
DISPLAY("Enter the location of the images to search: ")
images_to_search_location <- INPUT()
DISPLAY(“Do you want to blur faces that match or delete images containg matching
faces (TRUE=blur, FALSE=delete)? “)
blur <- INPUT()

face_to_search_for = face_recognition.load(face_to_search_for_location)
face_to_search_for_encoding = face_recognition.face_encoding(face_to_search_for)

arrImages_to_search = OS.LISTDIR()
int n=0
```

```
WHILE n <= LEN(arrImages_to_search)
    image = face_recognition.load(arrImages_to_search[n])
    image_encoding = face_recognition.face_encoding(image)
    results = face_recognition.compare_faces(face_to_search_for_encoding,
image_encoding)
    IF results == TRUE THEN
        IF blur == TURE THEN
            face_location =
face_recongnition.face_location(face_to_search_for_encoding)
            open_cv.blur(face_location, arrImages_to_search[n], write=TRUE)
        ELSE
            OS.RM(arrImages_to_search[n])
    ELSE
        PASS
    ENDIF
    n = n + 1
DISPLAY(“Finished”)

END Automatic_removal_of_persons_face_tool
```

Algorithm flow chart

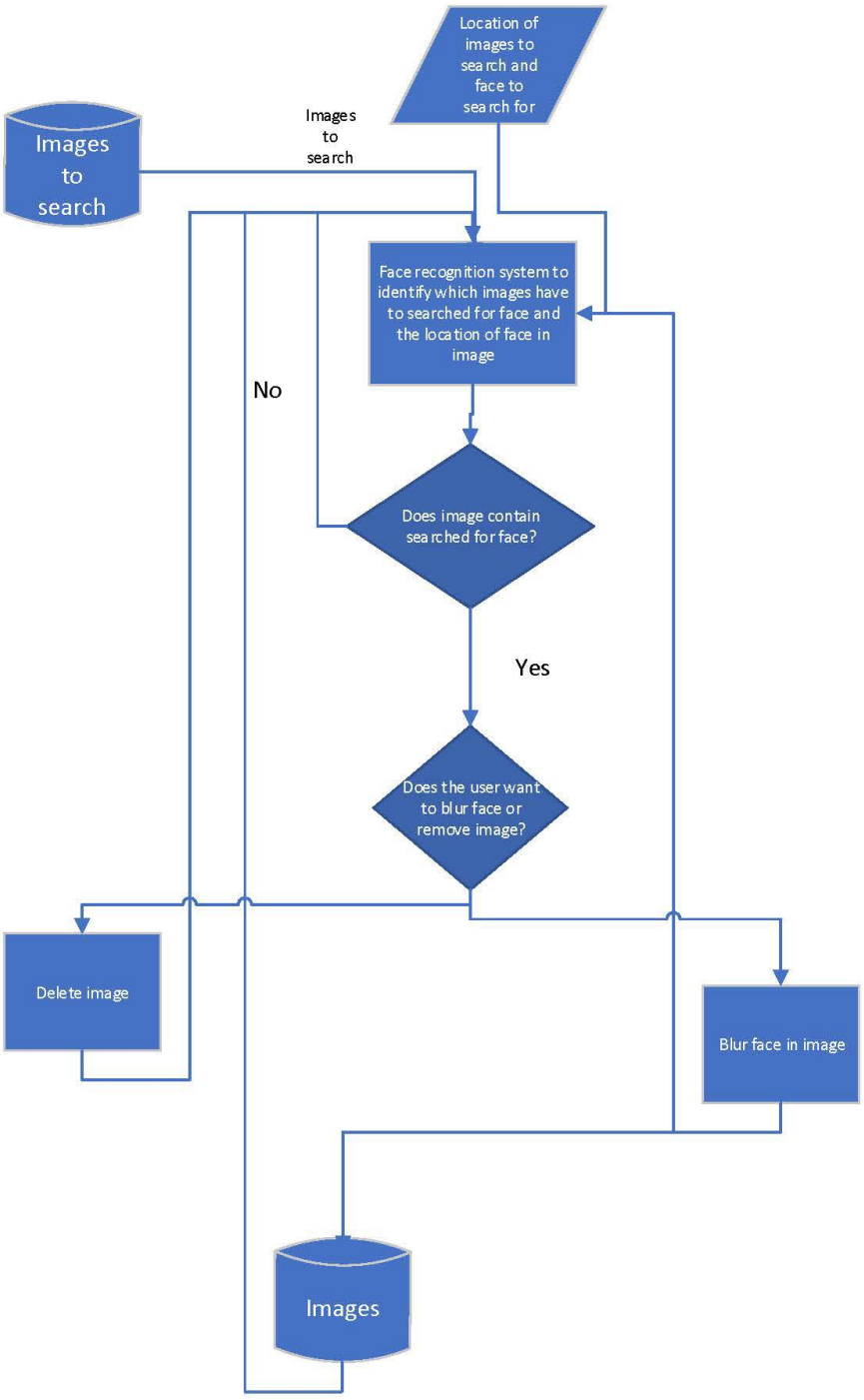


Refined system modeling tools

Data Dictionary

item	Data type	Description	validation
face_to_search_for_location	string	String that stores the location of the image containing the face to search for	File exist, valid file directory format
images_to_search_location	string	String that stores the directory of images to search	Directory exist, valid file directory format
n	int	Int to keep track of what image is being used	NA
face_to_search_for	numpy array	Face to search for loaded into face_recognition	NA
image	numpy array	Image to be search loaded into face_recognition	NA
face_to_search_for_encoding	numpy array	Variable to store the processed face to search for	NA
image_encoding	numpy array	Variable to store the processed image to search	NA
blur	Boolean	True/false value to specify if the program should delete the image or blur the face	Checks it is a valid Boolean

System Flow chart



Software environment

As this program will be doing advanced AI processing and image manipulation this makes many lower-level languages such as C or COBOL unsuited for this purpose. Out of high-level languages the best suited in my opinion is Python as it has support for packages and is the most popular for machine learning, allowing me to utilize per-existing packages to reduce the need for me to write extremely complex code. Python also has other benefits such as being an interpretation language, so I don't need to compile code when I am writing and debugging my software solution and its cross-platform support. The next best language for this project is C# because like python it is a high-level language that has machine learning capabilities; however, I didn't choose it because it doesn't have as good cross platform support which would make development much harder for me as a will be writing this software on a Linux machine and it would likely be run on Windows computers. Additionally, because it was created by Microsoft, I expect I'll have issues getting it to work on Linux.

Hardware requirements

This software will run on most x86_64 computers from the last 15 years. However, it may not run quickly. I would recommend using at least a quad core 3rd gen intel core series or newer to get decent performance. A GPU is not required however running the face recognition on a GPU will make it quicker. For using GPU acceleration, a Nvidia GPU with CUDA is required.

Data files

This program will only need the .py file to run or an equivalent file containing the software's code. The only other files needed are the image file the user input into the program.

Identifying Relevant Standard or Common Modules or Subroutines

This program is already reasonable modular due to the use of per-existing libraries, however the code to blur or remove the image could be split off to allow easier addition of different actions the program could take such as placing a new image over the identified face.

Social and ethical issues & Communication with others

In order to keep this document as ordered and logical as possible I have integrated these sections with the previous section on social and ethical issues and communication with others respectively.

User feedback

Depending on the final form of the software, it may be a web app or run locally, different systems will have to be in place to interoperate user feedback and updates. For both there will be links to a website and an email address likely at the bottom of the screen for the user to provide feedback via. If it is locally run there will have to be an additional module to check for and download updates if available to implement any changes suggested.

Documentation of design

I will be using Libre Office writer to maintain a logbook of my work and Libre Office cal (excel alternative) to manage a Gannt chart to plan my time. I will additionally be using GitHub for management of files and version control. This is in addition to updating parts of this document such as the data dictionary as required.

Gannt Chart:

Software year 12 task 1 logbook
Noah Hallows

Entry 1 29/10/23

Aim: My aim today was to make an official start to my project and set up the libraries I am going to use in the major project.

Work completed:

- Started Gannt chart.
- Started logbook.
- setup OpenCV, distrobox and “face-recognition” (a python package for face recognition)

Work to be continued:
I plan to continue updating the software I have installed and my logbook and Gannt chart.

Issues:
OpenCV doesn’t provide an install guide for Arch Linux, so instead of working it out myself I have set up a Fedora Linux container.

Entry 2 31/10/23

Aim: My aim today was to get a significant amount of my defining and understanding section done

Work completed:

- DFD
- IPO diagram
- Identification of the problem
- Social and ethical issues

Work to be continued:

- Structure chart

Issues: I had difficulty finding a program to make my diagrams in

Entry 3 1/11/23

Aim: To complete my context diagram

Work complete:

- Context diagram

Work to be continued:

- NA

Issues: NA

Entry 4 2/11/23

Aim: To finish my system modeling tools

Work completed:

- Structure chart
- System flow chart
- Data dictionary

- Generation of ideas section
- Communication with others section
- Updated IPO diagram
- Selected appropriate data types.
- Draft interface design
- Setup GitHub repo to store my work.

Work to be continued:
- NA

Issues: Microsoft Visio is bad; I had a bit of difficulty setting up GitHub for the first time and for some reason the GitHub android app doesn't let my upload files. I tried to use 'Mockups' to make my draft UI but I quickly got annoyed with it and made the smart decision to make a static website using Dreamweaver and the Bootstrap framework as the mockup UI instead.

Entry 5 3/11/23

Aim: I planned on making my pseudocode and algorithm flowchart

Work complete:
- Pseudocode
- Algorithm flow chart
- Updated Gannt chart
- Updated IPO diagram

Work to be continued:
- NA

Issues: Once again, I do not like Microsoft Visio and I'm not 100% sure how I will tell opencv which face to blur

Entry 6 4/11/23

Aim: I plan on completing the remaining work for my assessment

Work complete:
- Selection of software environment
- Identification of appropriate hardware
- Defining files
- Refined system flow chart
- Refined data dictionary
- Identification of potential subroutines
- Identification of appropriate test data
- Planed enabling of user feedback and update system.
- Reconsideration of social and ethical issues
- More communication with others
- Management of project using tools
- Compiling all work into single document ready for submission

Work to be continued

- NA, project completed.

Issues: I encountered no major issues today, the libraries and language I am using have no clear minimum hardware requirements, so I had to work it out myself.

Test data

TRUST ME