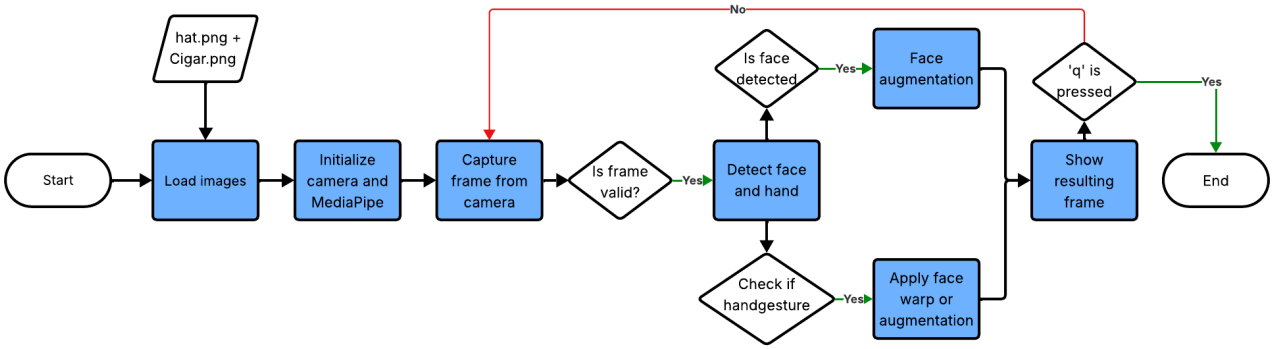
IPCV Project: Real-time Face Visual Effects
University of Twente
Version: 20250916v1

Final Report – Real-time Face Visual Effects

Group number: IPCV_Project_Group_21
Group members: Merel Hodes (2498863), Elke Putman (2624257), Annelies van Vastenhoven (3573966)

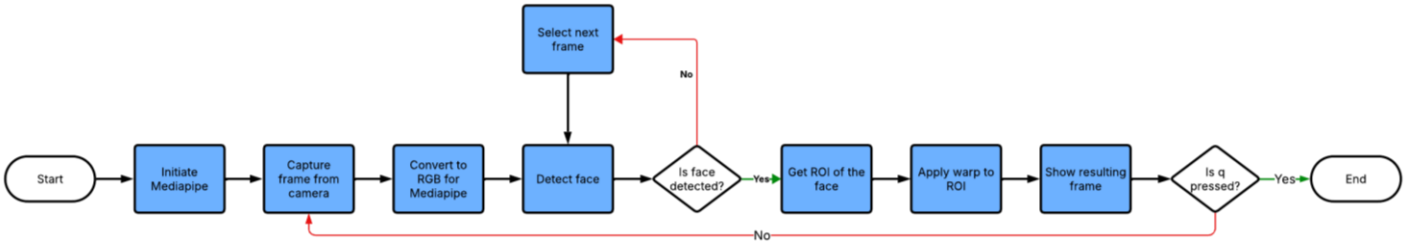
1 1. Basic system overview

1.1 Make an overview using flowchart to indicate how the Real-time face system constructed in software/development level.



The main pipeline initializes MediaPipe, a library with machine learning techniques to, for example, detect hands and faces.[1] The camera input is processed frame by frame. When a face is detected, a hat is placed on the head. Detected hand gesture cause different commands to be executed, and the updated frame is displayed in real time.

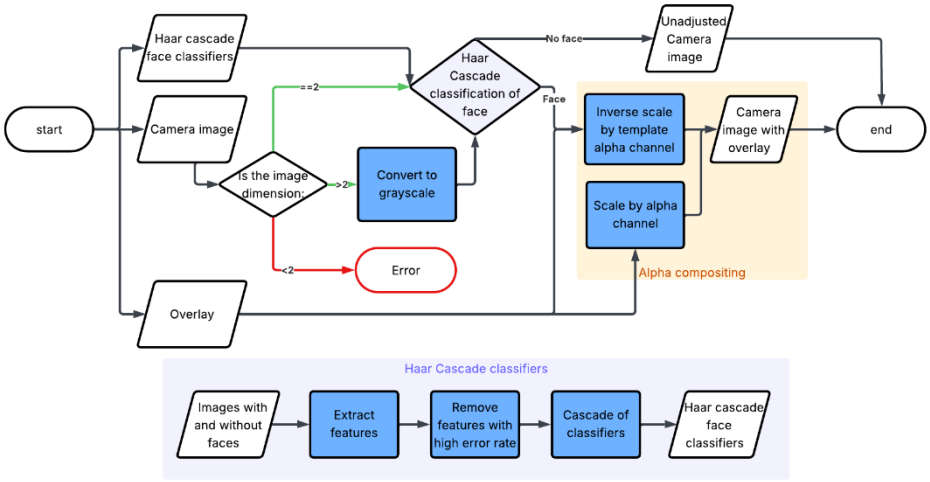
1.2 Technical roadmap for "Face Warp"



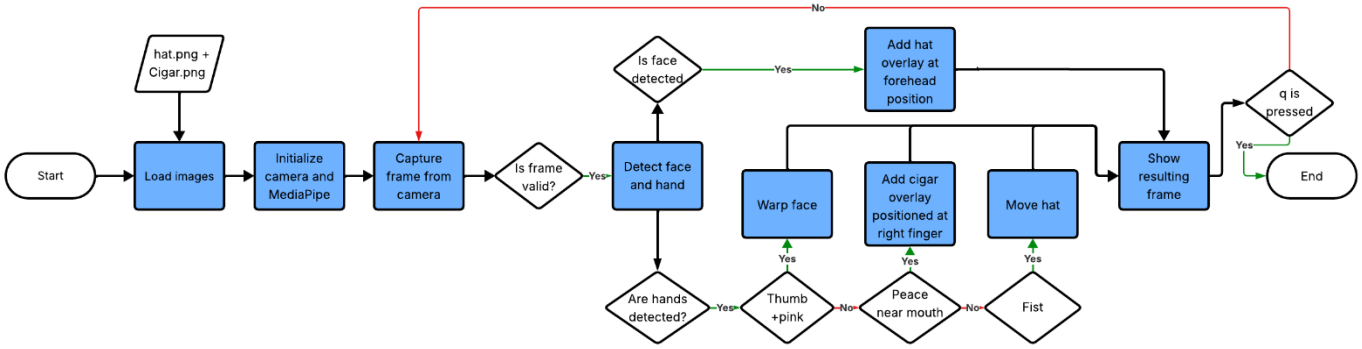
To start the face warp, the face is detected using Mediapipe. A region of interest (ROI) is then defined around the face so that the warp is only implemented on the face region, and moves consistently with facial motion. This ROI is determined using a relative bounding box, which calculates the position and size of the face as percentages of the total image width and height. The percentages are multiplied by the image dimensions to obtain pixel coordinates of the ROI. To implement the warp, a swirl effect is added. This swirl effect is centered on the ROI's midpoint, with a strength of 5 and a radius equal to the ROI divided by 1.3.

1.3 Technical roadmap for "Face Augmentation"

Face detection is performed using the Haar Cascade classifiers, these are built using an extensive library of images with and without faces to extract thousands of features. These features are then tested and reduced to the features with the best error rate, and to optimize performance a concept called 'Cascade of classifiers' is used. Cascade of classifiers divides the features into stages which check if a face is present – if there is no face at an early stage not all features have to be checked. Overlaying an image on top of the camera input is performed using alpha compositing. When no face is detected, the camera input is also the output.



1.4 Technical roadmap for "Motion Tracking & Interaction"



The motion tracking code uses MediaPipe to detect both hand and face landmarks. Hand gestures, such as fists, peace signs, and thumb/pinky extensions, are recognized through MediaPipe Hands combined with custom functions. Facial landmarks are detected with MediaPipe FaceMesh, identifying the face and mouth regions. For each frame, the code first detects faces and hands, then evaluates hand gestures. Depending on the gesture, it applies effects such as moving a hat, warping the face with a swirl, or adding a cigar overlay when fingers overlap the mouth.

2 Open Questions (answers can be short, but need to be understandable)

- Were there any challenges in working with real-time video images?
Yes, the real-time video is unpredictable and variable in subject and environment. This meant that the algorithm had to be more robust than when using a pre-shot video. MediaPipe hands and Facemesh are relatively robust algorithms [1], working well on different persons. However, presence of backlight, caused some troubles with detecting landmarks, thus making the code not entirely robust in all environments.

- What trade-offs did you encounter between algorithm complexity and running time in live demo?
To integrate hand motion tracking with face augmentation effects, the face detection method used in the face augmentation and face-warp algorithm was replaced with the MediaPipe Face Mesh algorithm. This approach eliminates the need to detect the face twice, allowing the facial landmarks to be used for both face augmentation and face-warping tasks. When implementing a face augmentation with a certain hand gesture, if the movement is too quick, the face augmentation has some delay in appearance

- How does the anatomical landmarks tracking influence your final performance?
The anatomical landmarks of the hands and face are used to position the hat, identify the mouth region, get a ROI for the face

and define various hand gestures. Overall, the detected landmark positions appear to be accurate. Their precise placement within each frame is crucial for the custom functions to correctly recognize and interpret the different hand gestures.

- How did you verify /evaluate / test your designed visual effects to make sure it is effective and robust?
We tested all aspects of the system under different circumstances.
To ensure robustness of the effects:

- The subject was varied in age, gender, and ethnicity.
- The environment was varied with measures of both artificial and natural light, including their orientation and brightness.

To ensure robustness of the program:

- Multiple devices were used

- how to prevent program bug/crash/clash/failure in live demo? What did you do?
Failure of the program would include the following problems, which we tried to account for:

- Environment preventing visual effects. → We researched the location of the demo; and ensured that the program would work in similar circumstances
- Program not running → Multiple devices were used to run the code, including the standard laptop provided by the University of Twente
- Facial recognition not working → We tested our visual effects on different ethnicities and genders.

- Which part of the workflow was most informative or surprising to you?
Most informative was the effect of the backlight on the face detection. Mediapipe is quite a robust algorithm but still experienced some issues in this regard. Furthermore, the fact that a lot of libraries are already available for use and how to use them was very interesting. However, we found it surprising that operations with an alpha channel is not automatically included in open-cv libraries.

3 4. Conclusion and Reflection

In this project, several Python libraries were utilized to detect hand and face landmarks. These landmarks were successfully used to define hand gesture-based interactions and determine the location for overlays or a face warp. The performance was overall robust over different subjects or environments, with some small issues with backlighting and fast movement.

5. References

[1] Zhang F, Bazarevsky V, Vakunov A, Tkachenka A, Sung G, Chang C-L, Grundmann M. MediaPipe Hands: On-device real-time hand tracking [Internet]. arXiv preprint arXiv:2006.10214. 2020. Available from: https://arxiv.org/abs/2006.10214