Team 18

Agenda

- Requirements & Implementations
  - Hair Style
  - Hair Color
  - Salon Recommendation
- **Demo**
- Validation and Test
- Future Implementation Plan
- Q&A

Hair Style

- FR1: The user must input their face in image or video format
- HM5: The user must choose a hairstyle from the provided options if hair style transformation is selected
- HM7: the system should adjust the coordinate to be proportional with the image scale
- HM8: The system should add the new hairstyle coordinates above the facial coordinates
- HM9: The system should output the transformed face in image or video.

- FR3: The system must perform face shape detection on facial image
- FR6: The system must mark the corners and shape with facial landmark coordinates

Implementation:

- GLTF models were first converted to JavaScript files using *gltfjsx*, then loaded using *@react-three/drei*
- *Three.js*, *@react-three/fiber*, *@react-three/drei* are used to manage canvas used for displaying AR
- *jeelizFaceFilter* is used for facial tracking and WebGL computations

Hair Color

- FR1: The user must input their face in image or video format
- HM1: The user must choose a color from the provided options if hair color transformation is selected
- HM3: The system should map the hair coordinates on the image and apply the new rgb values to the region
- HM9: The system should output the transformed face in image or video.

- FR2: The system must pre-process the image by transforming the input facial image to grayscale
- FR3: The system must perform face shape detection on grayscale facial image
- FR6: The system must mark the corners and shape with facial landmark coordinates

Implementation:

- Socket.io and flask are used to enable bidirectional communication between backend and frontend.
- Backend is responsible for hair segmentation and hair color change.
- Frontend would display the changed hair color through image or video streaming.

Salon Recommendation

- HR1: The user must enter a address or turn on current location to use the recommendation system
- HR2: The system must retrieve nearby hair salon locations based on the input address
- HR3: The system should label the retrieved locations on a map
- HR4: The system should rank the locations

Implementation:

- Integrated with Google Places API to search for nearby interests and place details
- Used Bayesian Average algorithm for ranking the salons based on the ratings and number of reviews
- Established an server endpoint for the client to retrieve information
- Used Google Maps API to display the map

Recent Changes

- IOS application -> Web application
- The AR hair model simulation only allows live video as input.

Demo

Validation and Test

- Functional Test
  - Implementing unit test
    - Backend Test
      - Postman unit test for the flask service
      - Pytest unit test for image processing
      - Jest unit test for the AR component
    - Frontend Test
      - Jest unit test for frontend UI functions
  - Integration Test according to VnV plan
- Non-functional Test
  - Latency Test
  - User test
    - Appearance
    - Ease of Use
    - Other non-functional tests listed in VnV plan

Revision 0 to Revision 1

- HairColor Module (Implementation)
  - Alpha value slider* (UI & backend)
  - Download an image or a snapshot of the live video* (UI & backend)
  - Potentially improve model performance* (backend)
  - Color switch delay (have to press twice to see the hair color change)* (UI & backend)
- HairStyle Module (Implementation)
  - Change color of hairstyle model at runtime (UI & backend)
  - Provide more & finer hairstyle models (UI & backend)
  - Improve performance

Revision 0 to Revision 1

- Salon Recommendation Module (Implementation)
  - Add some sort of loading indicators while fetching data from the server
  - Map flicker fix
- Testing
- Deployment
  - Deploy to host, backend requires virtual machine
  - Meet processing power requirements
- Documentation

Q & A