FaceQuery 2.0 – Making Faces Explorable

Problem Statement

In my previous project (Team 156, Fall 2022), we managed to build facial query application which allows user to query and compare uploaded faces with the faces of that in the data source. A peek into previous project's application can be seen in the Appendix: FaceQuery 1.0 section. We managed to go outside most of the modern-day applications of face recognition use cases, which are mainly in security and access control applications (Face Recognition Methods & Applications, 2013).

We validated our result by randomly sampling several person identity pictures in the data source, and using the average feature representations of each identity, we compared them to known positive and negative matches based on cosine similarity, although we observe similar performance metrics with the state-of-the-art at that time using pictures within the same dataset for validation and testing, the performance of comparing to outside of data source face pictures (via image upload), the resemblance were not really that good, most probably because the input pictures extraction is limited to 31 attributes of interest out of 40 available attributes, which might not be enough to get uniqueness of faces.

In addition to that, the functionality of querying dataset is limited to one shot query of the dataset without going deeper in the result, be it from the attributes query or the image upload query, this means that users can't go dig down further based on the results from previous queries. The attributes-based search was also inclusive only, in that, there's no possibility to query *attractive* and *young* but ***not*** *bald* attributes, as it doesn't support exclusion/negation or notion of not having an attribute. Another missing feature is that the similarity between query results was not being explored further, which is important in the case of finding individual(s) of interest.

Data Source

Using the same dataset as (Team 156, Fall 2022) this project will be using CelebA - CelebFaces Attributes Dataset (Deep Learning Face Attributes in the Wild, 2015), which is a large-scale face attributes dataset with 202,599 celebrity images, each with 40 attribute annotations from 10,777 individuals. Which is available at following link: https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Methodology

For this project, 39 attributes of interest (excluding *5_o_Clock_Shadow* attribute) out of 40 available attributes will be used as compared to previous project (Team 156, Fall 2022) which used only 31 attributes. Due to this implementation decision, all codes for this project are built from scratch since average of attributes of interest for all images representing an individual must be recalculated, then cosine-similarity-based match percentage:

$$Match\ Percentage = \frac{\vec{a}.\vec{b}}{||\vec{a}||\,||\vec{b}||} \times 100\%$$

of the average attributes of interest for each different pair of 2 individuals also needs to be recalculated.

In the case of image-upload-based query, there are two methods used, first, the 39 attributes of interest are extracted from the input picture(s) using variation of (WeblineIndia) facial feature extraction implementation and are filled automatically to the attribute-based-query selection, which then filter images in the database based on the attributes selection, second, with the limitation of 39 attributes, although for some cases the result somewhat resembles the input picture, but there were big room for improvements, I've conducted qualitative test with pictures of my family members, and they don't seem to agree with the result indicated by loud laugh after seeing their result. Thus, I implemented variation of (Esler) to encode face in 512 attributes vector, sacrificing interpretation of face attributes for hopefully higher resemblance with image input.

For the case of (Esler) implementation, each of the image in the dataset has been processed, resulting in 202,305 successfully extracted vectors out of 202,599 available images in the dataset. Fairer validation from the one conducted in (Team 156, Fall 2022) will also be pursued for quantitative comparison.

The user interface is getting revamped as well, to overcome the problem of inclusive only query. Now, the users can query attribute's irrelevance, inclusion, or exclusion. Matching individuals can also be added on top of previous query result, not limited by maximum of 100 individuals in the visualization.

The query result network is now represented in a way that encodes the cosine similarity between two individuals in the query result, making it more feasible to use in the case of finding individual(s) of interest.

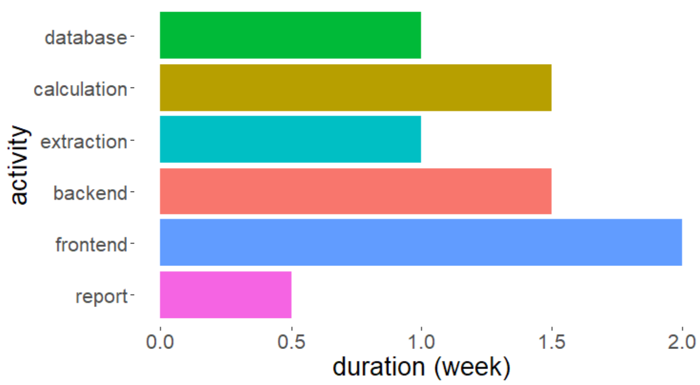

Figure 1: Project activity breakdown

Figure 1 shows the activity breakdown distribution of 7.5 weeks project duration:

1. Database

- Design and creation of database.

2. Calculation
   - Recalculation of match percentages between 10,177 entities resulting in $\binom{10,177}{2} = 51,780,576$ combination.
   - Including evaluation calculations.
3. Extraction
   - Extraction of 512 attributes from 202,599 available images.
4. Backend
   - REST API creation for database access and similarity calculations.
5. Frontend
   - Design and implement UI and visualization.
6. Report
   - Writing of this report.

Database	Calculation, Extraction, Backend	Frontend
SQLite	PyTorch, RAY, Flask, torch-directml	BACKBONE.JS, jQuery

Evaluation and Final Result

To showcase the current result, we will go through steps to find individual of interest using outside of the dataset picture, my selfie:

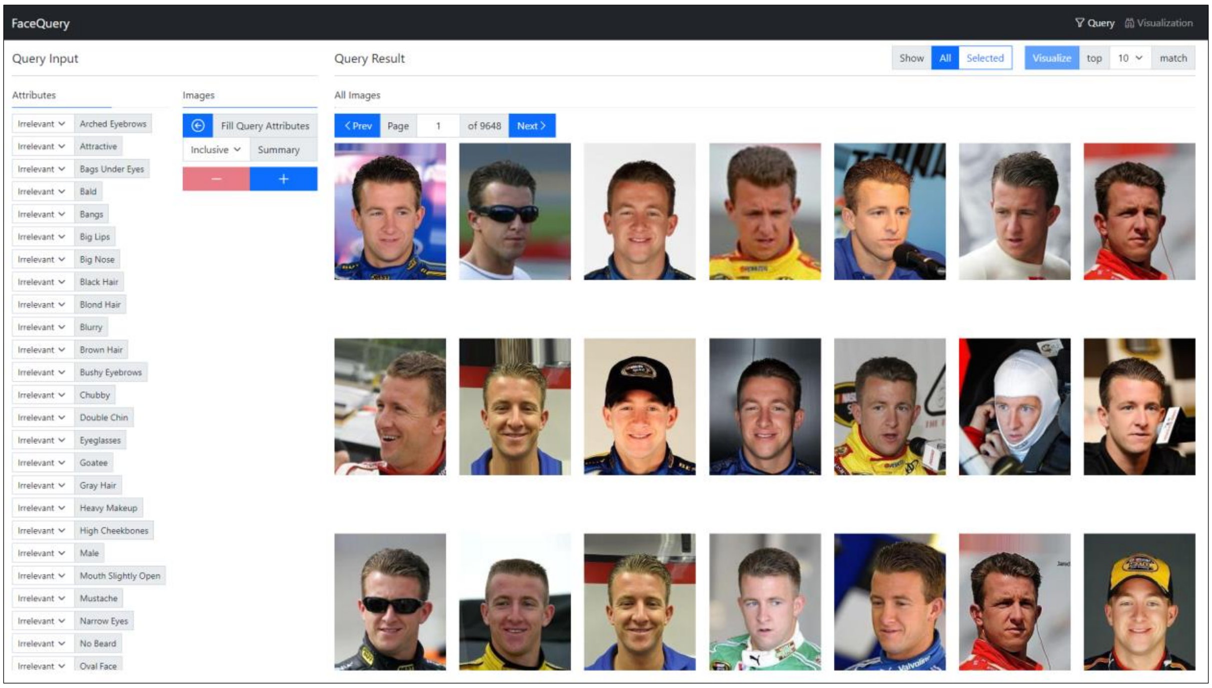

Figure 2: FaceQuery 2.0 Initial Page

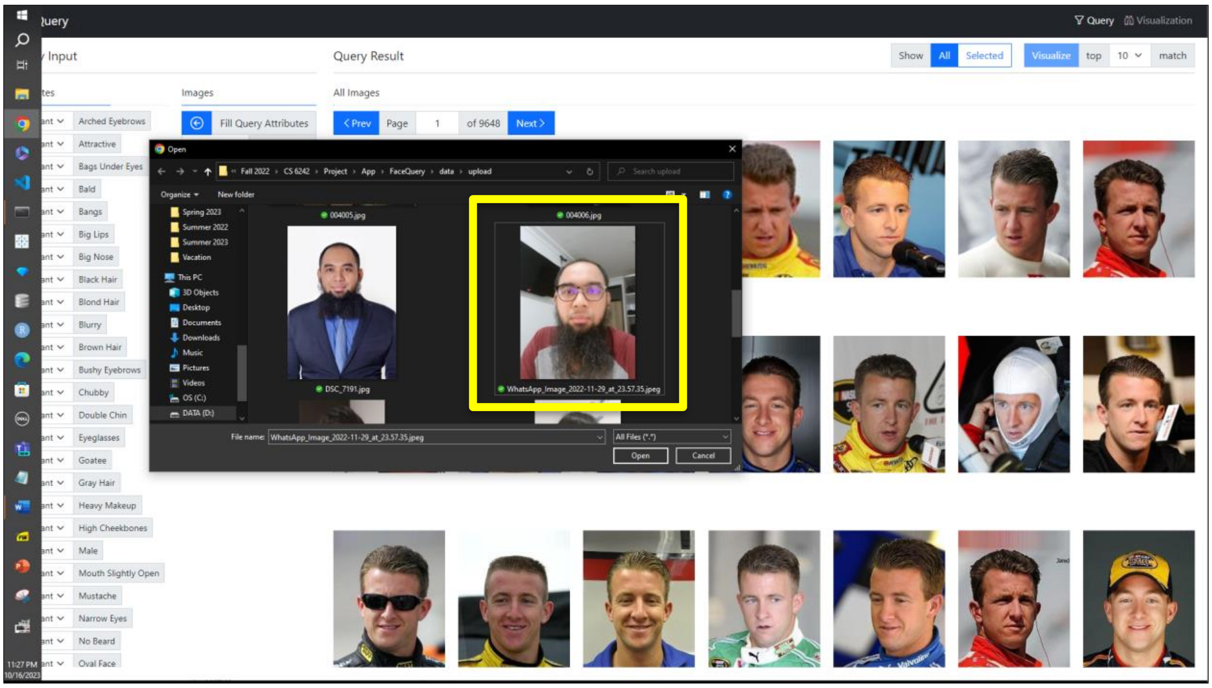

Figure 3: Selecting image input.

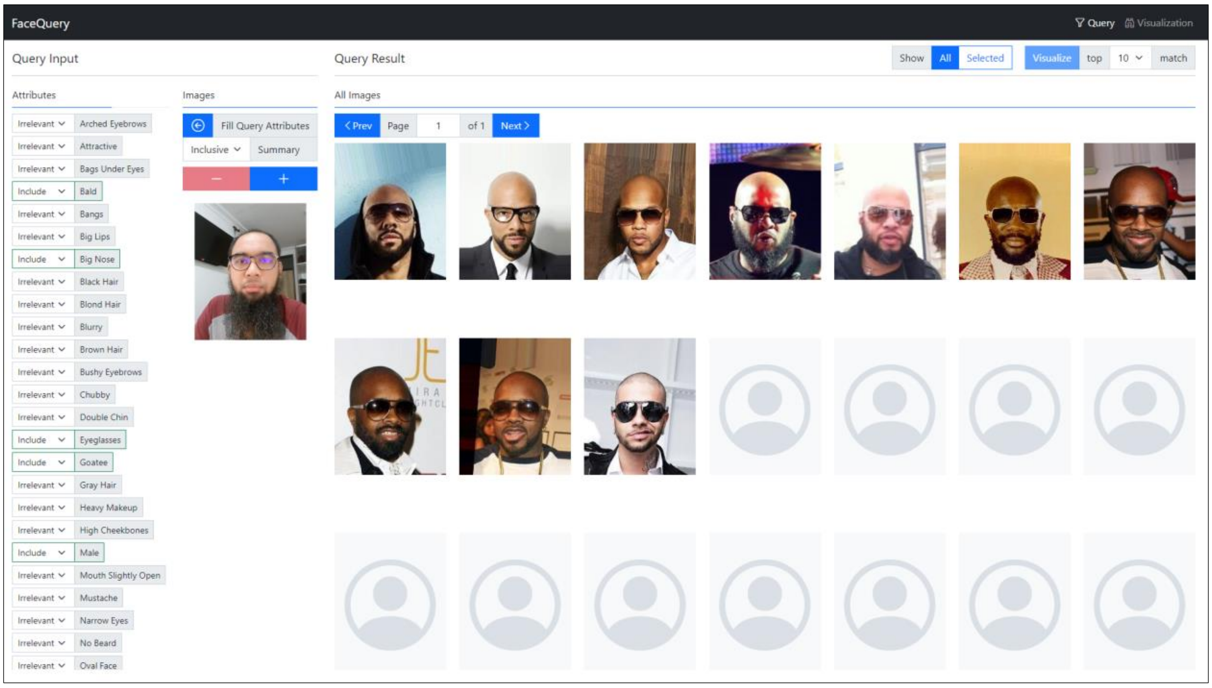

Figure 4: Image based query result.

It seems the result pretty much resembles the input image. We can now do exclusion query, let's exclude *Eyeglasses* attribute:

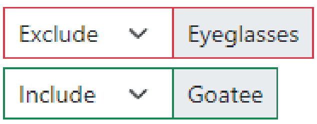

Figure 5: Exclude *Eyeglasses* attribute.

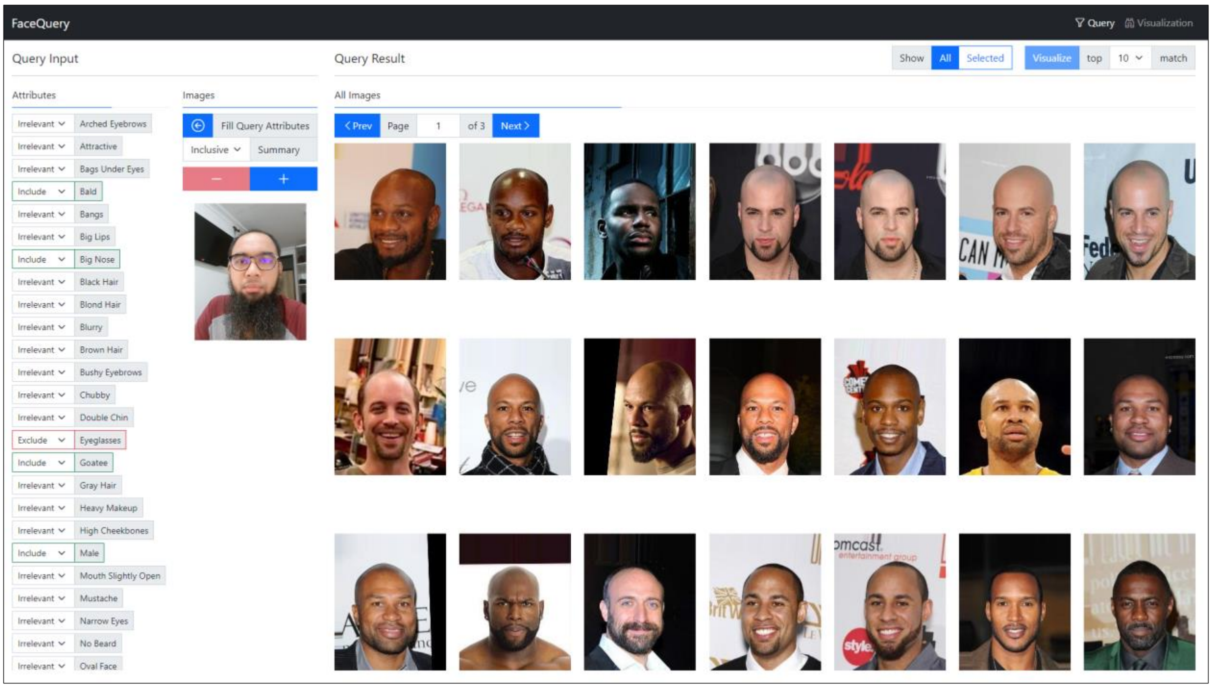

Figure 6: Result after excluding *Eyeglasses* attribute.

The exclusion seems to be working. Now let's include *Eyeglasses* attribute back, then select several pictures that we think closely resemble input picture.

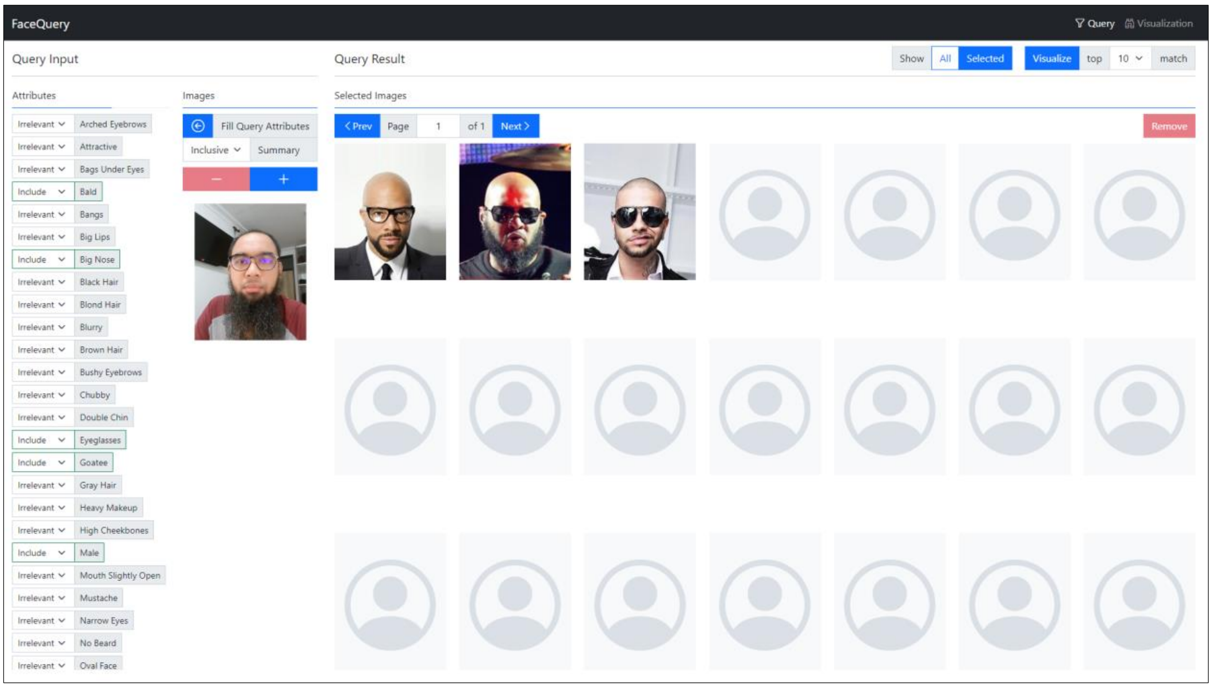

Figure 7: Select 3 closest pictures.

The selected pictures in Figure *7* come from query result selection in Figure *4*. In Figure 8, we are visualizing 10 highest matches from 3 selected pictures in Figure 7.

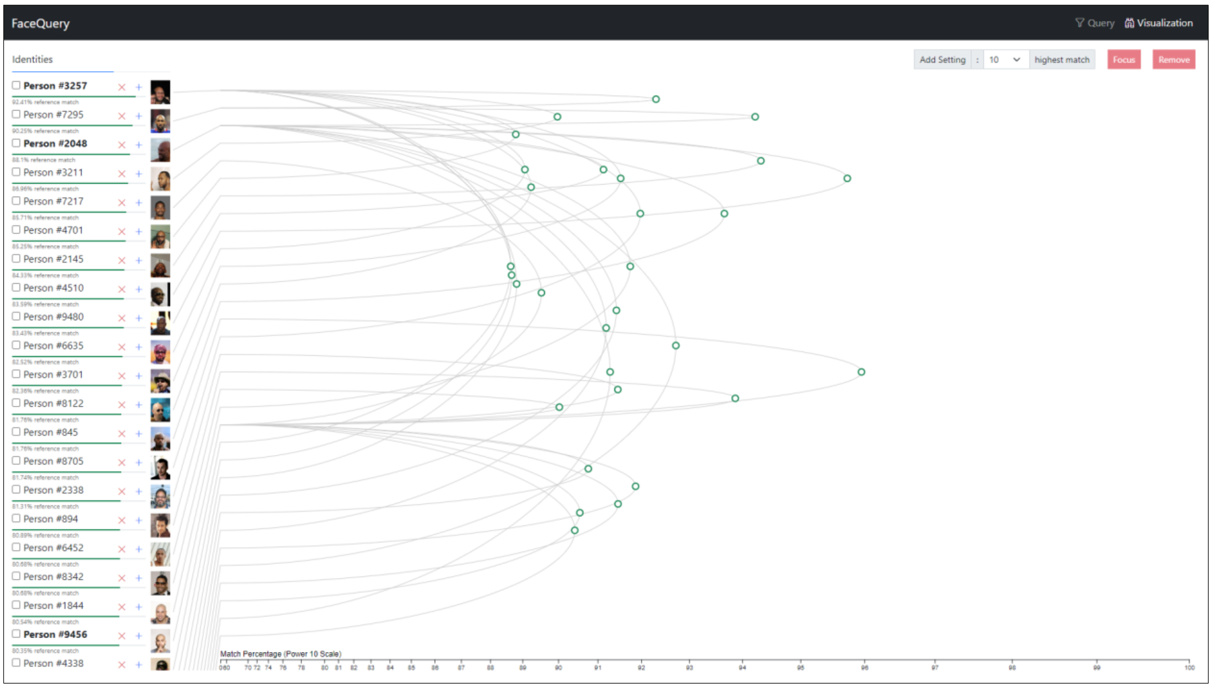

Figure 8: Visualize 10 highest match from selected 3 individuals.

Each person identity has their representative picture thumbnail with their respective match percentage from average identity attributes of selected pictures in Figure 7. We can also see the utilization of match percentage value in the connection between the 10 highest match individuals from selected pictures.

Clicking the thumbnail will display the comparison page comparing input pictures and previously selected pictures with all available person identity pictures in the system for easier cross reference check:

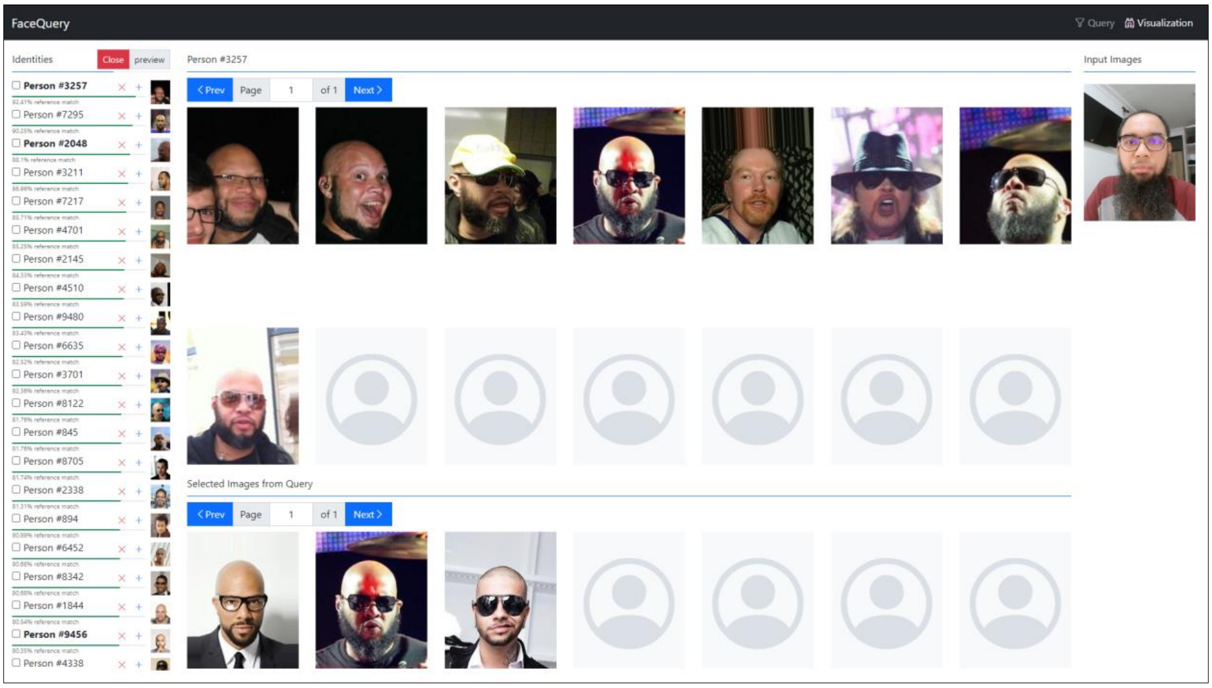

Figure 9: Cross reference check.

Selection in Figure *10* is done via brushing action, as for Figure *11*, the network visualization with match percentage value can be used as filter, as well as the individual label checkbox.

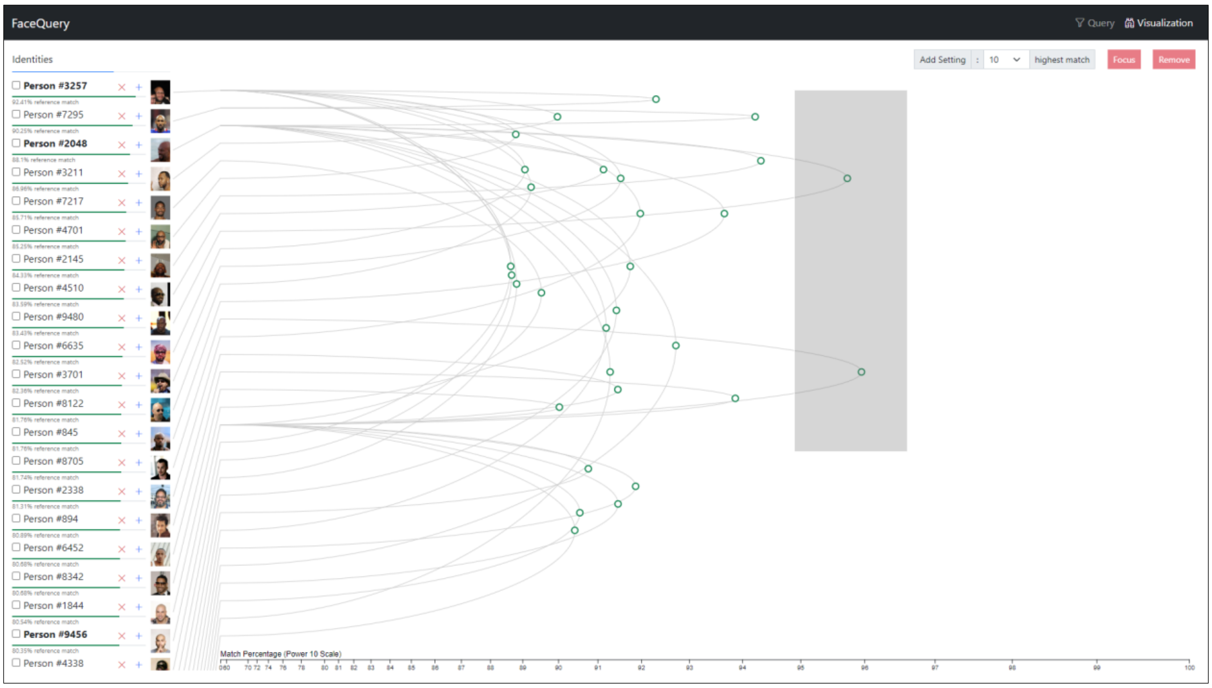

Figure 10: Selecting 2 highest match percentage between individuals.

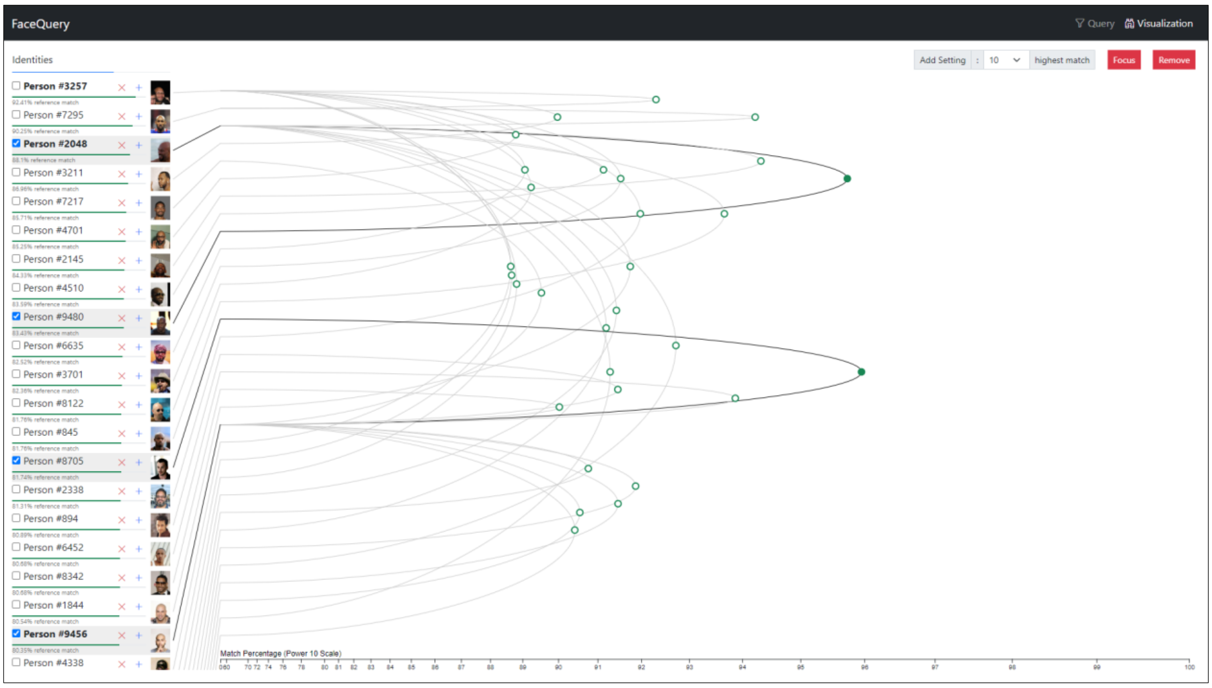

Figure 11: Selection result.

We can then either focus or remove the selection, the result of focusing on selected individuals are shown in Figure *12*.

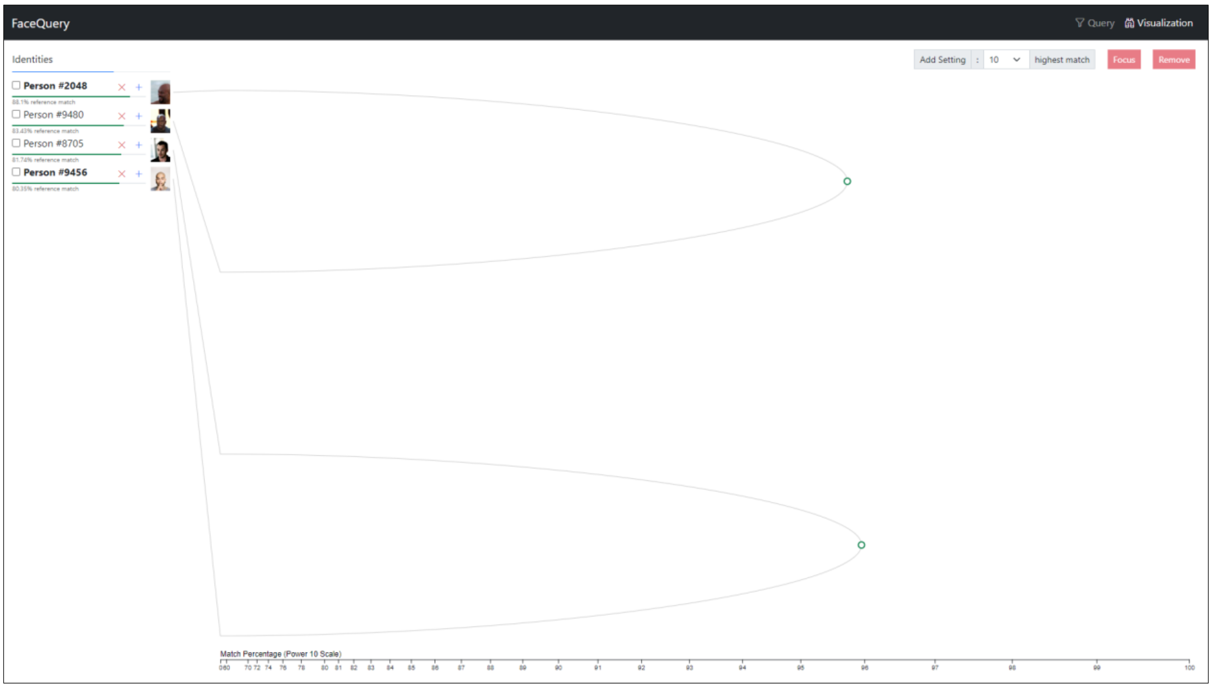

Figure 12: 4 Individuals with highest match percentage.

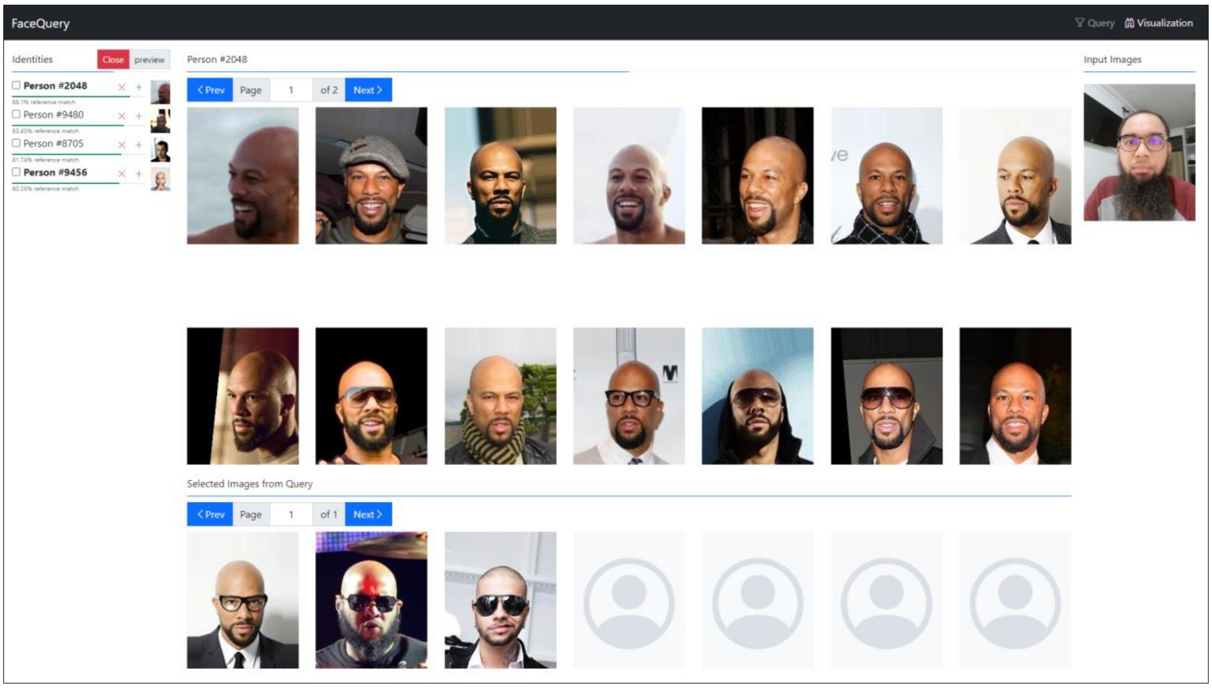

Figure 13: First of 4 highest match individuals.

Now we will discuss evaluation strategy. Quantitatively, evaluation in (Team 156, Fall 2022) had been done by comparing 1 picture of person identity in the database with the average feature representations of a randomly selected identity in the database. This one random identity has 0.5 probability that it come from the same identity and 0.5 probability that it come from different identity, then check the match percentage, after that apply some threshold that if it is $\geq 50\%$ match then the test picture belongs to the randomly selected identity, otherwise it doesn't belong to the randomly selected identity. This was done for 2000 repetitions. There are at least two problems I realized later with this strategy, first, this means that we are taking sample with maximum size of 2 in each of the repetition from the database, but in reality, we are comparing way much greater than 2 sample size since we have 10,177 identities in the database, second, the test picture is part of an identity's average feature representations, thus this is actually a data leak if we are comparing it with the average features of the same identity, thus a new evaluation strategy was designed with pseudo code as follows:

for i in $\{1,...,2000\}$:
  take m samples from 10,177 individuals in database
  take all pictures that belong to these m individual samples, let:
    n : total number of all pictures from m individual samples
    n^j: total number of pictures belong to an individual $j, \forall j \in \{1,\cdots,m\}$,
    thus we have $n = n^1 + \cdots + n^m$
  take 1 picture from total of n picture samples as test set
  calculate match percentage of the 1 picture test with the rest $n-1$ picture samples
  count average of $n-1$ match percentage values grouped by identity id
  get identity id that has highest match percentage average
  compare identity id that has highest match percentage average with the real identity id of the test picture
calculate the accuracy score

with this setup, there is no data leak as the validation data does not contain the test data, and we can have different sample sizes other than 2. Below is the result for sample size with $m \in \{2, 10, 100\}$:

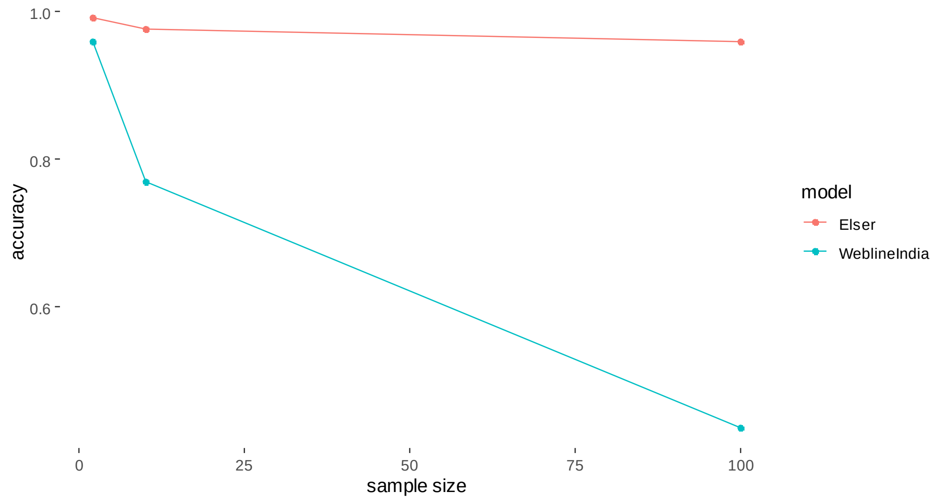

Figure 14: Quantitative evaluation result

we can see that as the sample size goes higher, the accuracy drops, and this drops significantly for (WeblineIndia) based model 0.9595, 0.769, 0.4355 while we have 0.9915, 0.9765, 0.959 for (Esler) based model for sample size of 2, 10, and 100 respectively. In Figure 15 and Figure 16 as an example for qualitative comparison, we can see that (Esler) based model give more accurate results compared to (WeblineIndia) based model:

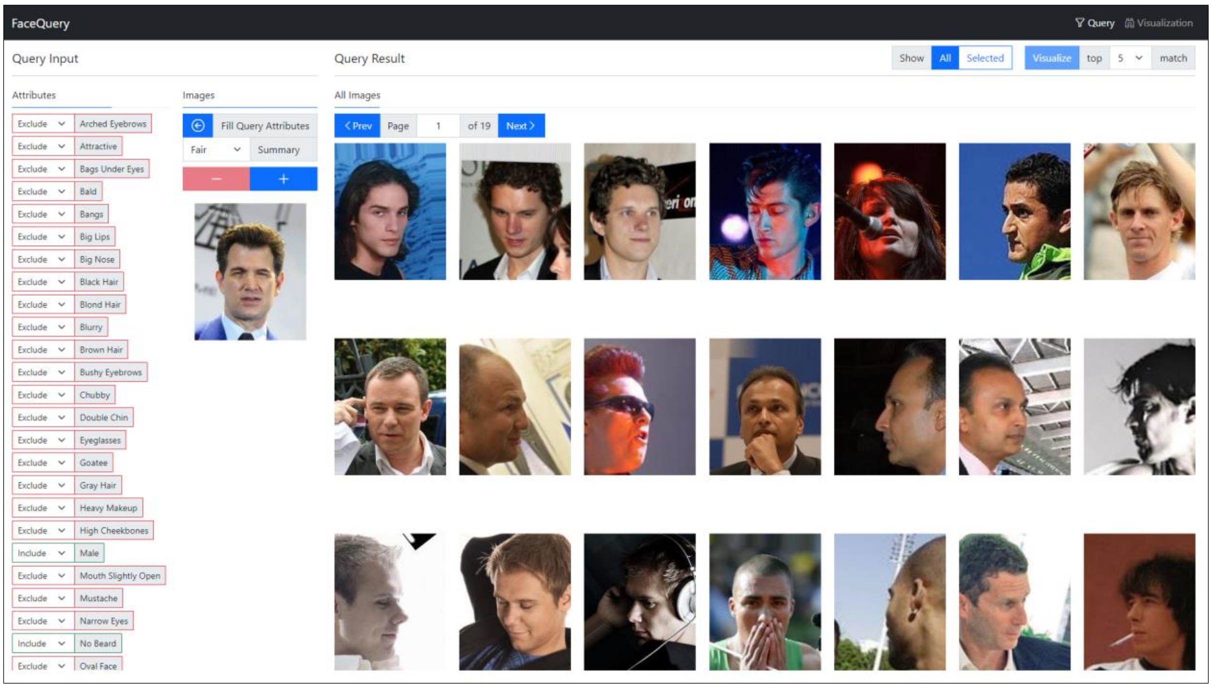

Figure 15: (WeblineIndia) based model result

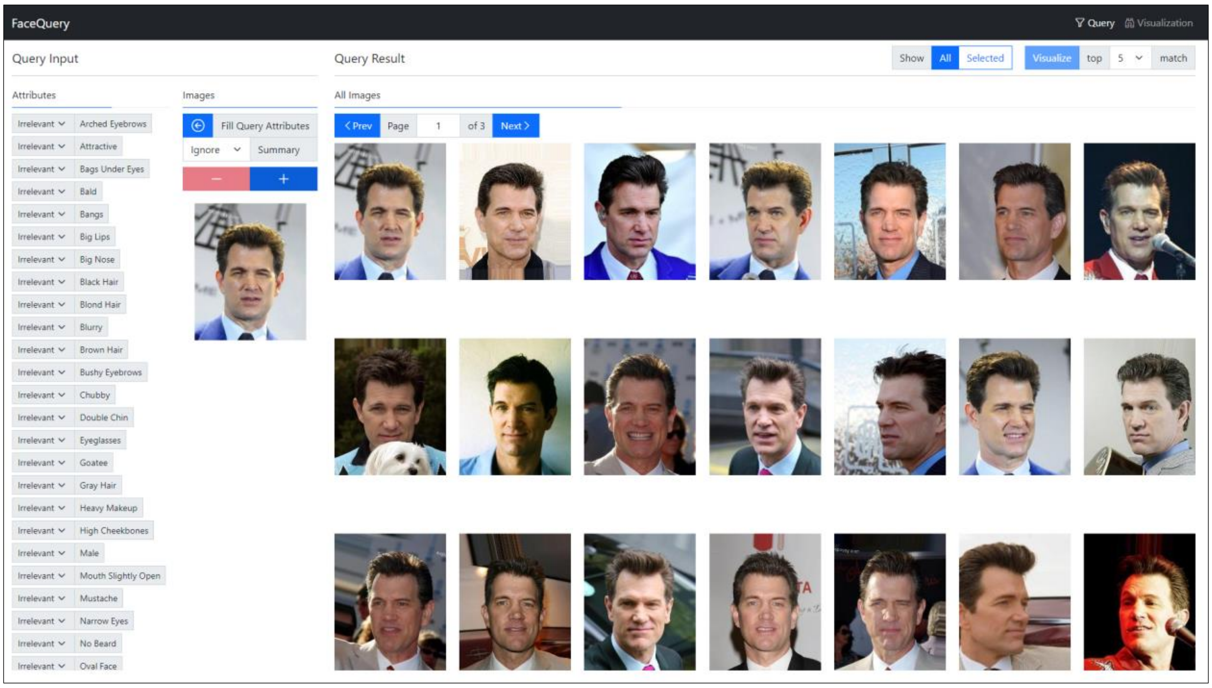

Figure 16: (Esler) based model result

Conclusion

In this project, we have tackled below problems:

1. Image-upload-based query accuracy.
2. Inclusive only and 100 maximum result query by extracted facial feature.
3. Data leak in quantitative evaluation.
4. Unutilized match percentage in network visualization.

by adopting below methods:

1. Implement a more accurate neural-network-based model.
2. Fix frontend and backend, explore all identities, not inclusive only query.
3. Separation of test and validation data.
4. Visualize network and match percentages, make it more explorable.

Appendix Face Query 1.0

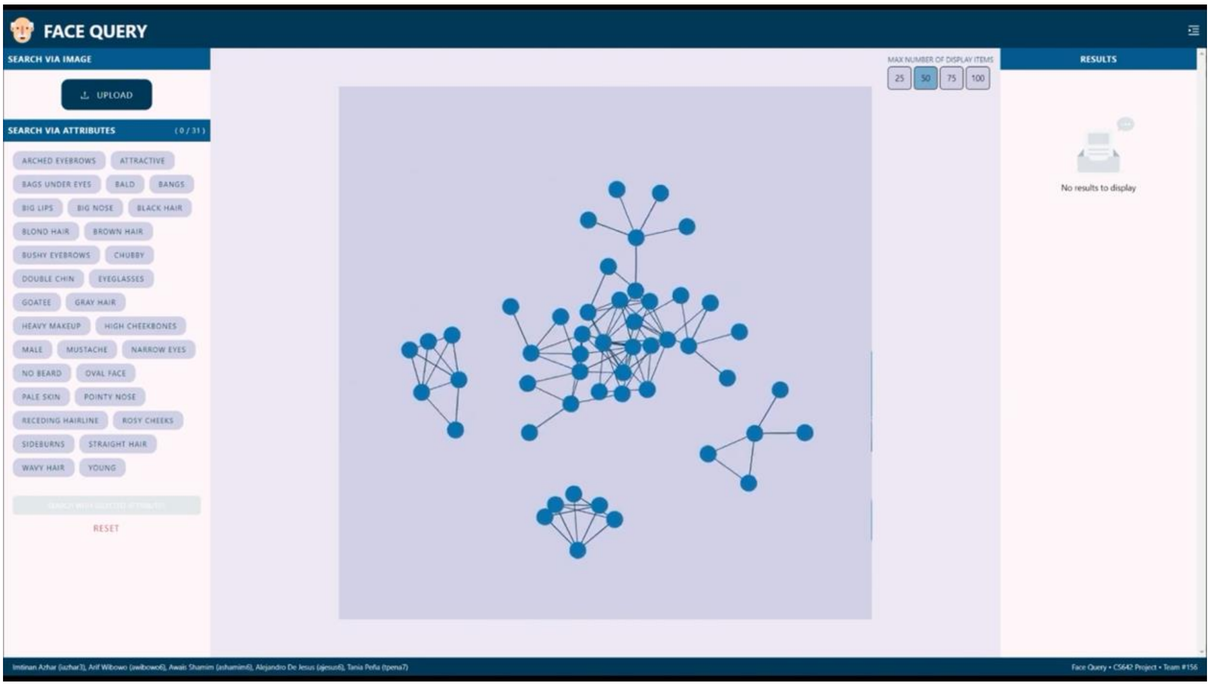

Figure 17: Initial Display

Figure 18: Query by attributes (male, no beard)

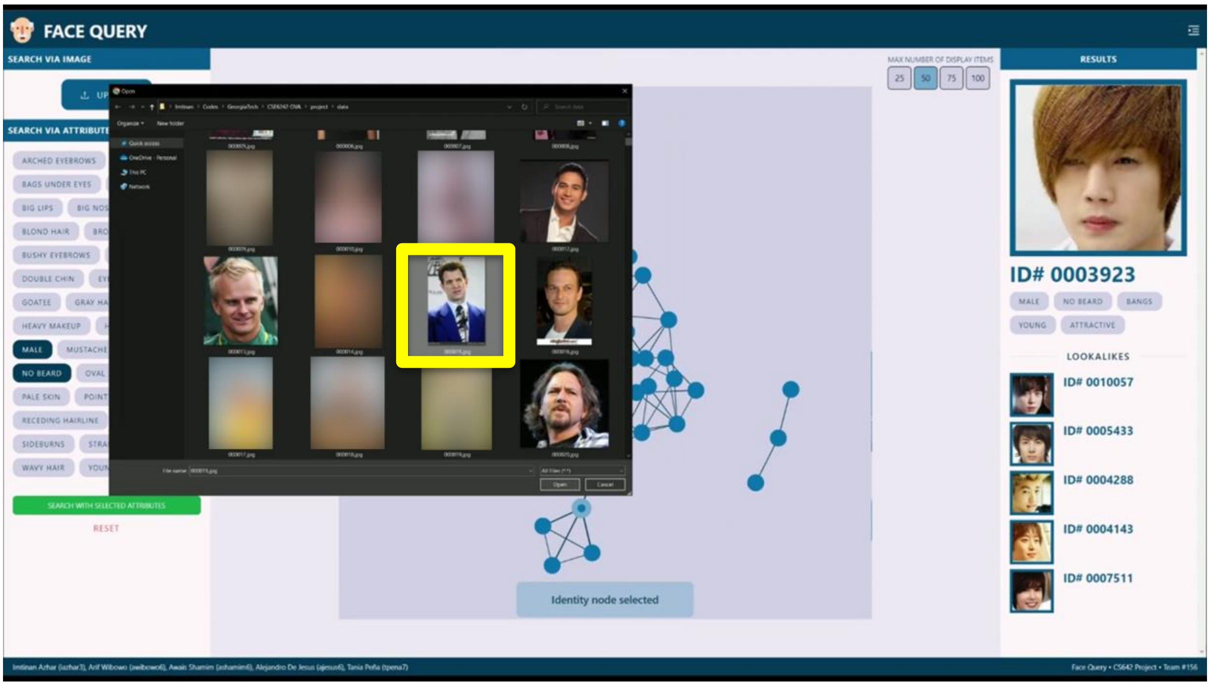

Figure 19: Query by image upload

Figure 20: Query result from image upload

References

Deep Learning Face Attributes in the Wild. **Liu, Ziwei and Luo, Ping and Wang, Xiaogang and Tang, Xiaoou. 2015.** 2015.

Esler, Tim. facenet-pytorch. [Online] https://github.com/timesler/facenet-pytorch.

Face Recognition Methods & Applications. **Parmar, D. N., & Mehta, B. B. 2013.** 2013, International Journal of Computer Technology & Applications, Vol. 4, pp. 84-86.

Team 156, 1. Adhitya Arif Wibowo, 2. Alejandro De Jesus, 3. Awais Shamim, 4. Imtinan Azhar, 5. Tania Pena. Fall 2022. *FaceQuery: Making Faces Queryable - CSE 6242 Final Report.* Fall 2022.

WeblineIndia. *AI ML - Human Attributes Detection with Facial Feature Extraction.* [Online] https://github.com/weblineindia/AIML-Human-Attributes-Detection-with-Facial-Feature-Extraction.