École Polytechnique de Louvain

LDATA2010 - Information Visualisation

Project - Visualization Dashboard

December 13, 2024

John Lee - Professor

Pierre Merveille - Assistant

Pierre Lambert - Assistant

Victor Joos de ter Beerst - Assistant

Mathis Delsart - 31302100

2024 - 2025

Chapter 1 - User Guide

0.1 Dataset Selection

I selected the `CelebA` dataset, which contains embeddings. As machine learning, particularly GANs and latent spaces, is one of my main areas of interest, I was keen to analyze these embeddings that represent celebrity images.

0.2 Home Page

The homepage introduces the visualization tool, explaining its purpose. Every 3 seconds, three images from the dataset are displayed. The page also features three buttons leading to other visualization sections and a navigation bar at the top of the screen for easy app navigation.

0.3 Data Analysis Page

This page presents various statistics of the dataset, including the number of images, embedding dimensions, mean values of facial attributes, and the count of images with $X\%$ or more of specific features (X being user-defined). It also includes five interactive graphs, each accompanied by a description and an explanation of its interpretation. The graphs are zoomable, and hovering over them reveals additional information.

- Correlation of the 39 facial features to identify linear relationships between features.
- Distribution of feature presence to check dataset balance.
- Distribution of embeddings to analyze embedding values.
- Co-occurrence matrix of features to explore relationships between them.
- Correlation between facial features and embedding values to understand which features tend to have positive or negative embedding values.

All of these plots are illustrated in Section 0.6.

0.4 Similarity Search Page

This page allows users to search for the most similar celebrity to any image selected from the dataset, which contains 30,012 images. Users can choose any image from the dataset, and based on the selected distance metric (e.g., cosine, Euclidean), the tool will find the i-th most similar image to the chosen one, where "i" can range from 0 to 30,011 and is user-defined.
If the user selects the top 20 most similar images, the results will include several images of the same celebrity, indicating they are close in the embedding space. This feature provides a detailed way to explore and compare images based on their embeddings.

0.5 Embeddings

The Embeddings page is the most complex, comprising three main components:

- `Feature Selection:` Users can select among 39 facial features. There is an option to select or deselect all features at once, or to adjust the weight of each individual feature. The application calculates a weighted average of the embeddings based on the user's feature selections and weightings, allowing exploration of the latent space. For example, increasing the weight for the "Bald" feature will show a more balding person. Since the machine learning model for generating images is not available, the tool plots the most similar image from the dataset based on the adjusted embeddings.
- `Dimensionality Reduction:` After adjusting the weights, users can apply five dimensionality reduction algorithms to visualize the reduced 2D embedding space. The space is filtered according to the user's selected features (only features with weights greater than 0 are considered). Additionally, users can toggle between an "AND" or "OR" filter. With "AND" activated, only images containing all selected features are shown; with "OR," images with at least one selected feature are displayed. The filtered embedding space is then visualized in a 2D plot. The number of images remaining in the filtered dataset is also displayed, providing insight into the size of the subset being visualized.

- `Clustering`: Users can apply clustering algorithms to the reduced embedding space to visualize different clusters. Three clustering algorithms are available, and their results are shown graphically. As users hover over points on the graph, the corresponding image is shown, enabling a more intuitive understanding of the embedding space.

0.6 Visualizations of Facial Feature Analysis

This section presents visualizations that explore various aspects of facial features and their relationships with embeddings. The following plots provide insights into correlations, co-occurrence, and distributions of features, as well as their relationship with embedding values.

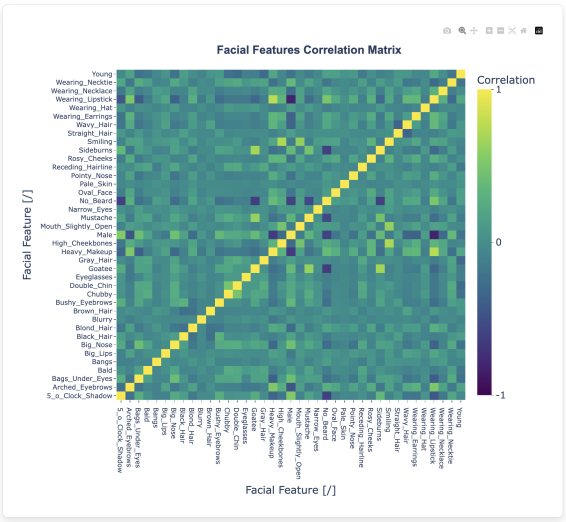

Figure 1: Correlation Matrix of Facial Features

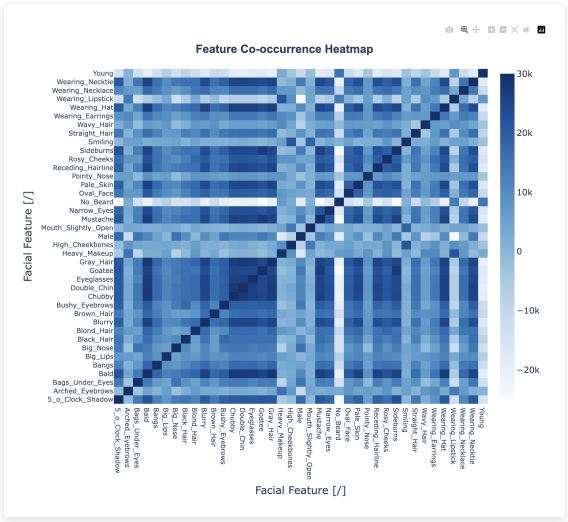

Figure 2: Co-occurrence Matrix of Facial Features

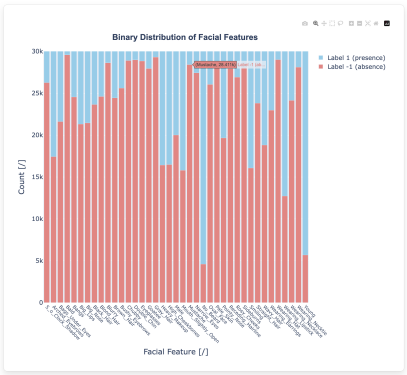

Figure 3: Distribution of Facial Feature Presence

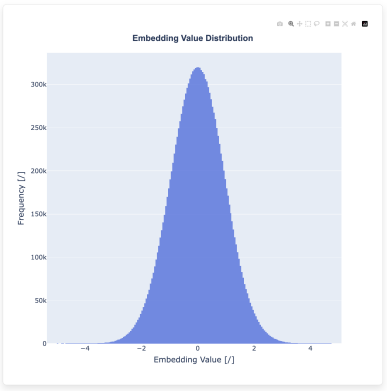

Figure 4: Distribution of Embedding Values

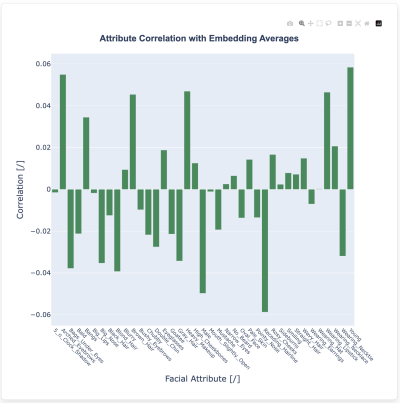

Figure 5: Correlation Between Facial Features and Embedding Values

Chapter 2 - Design Choices

0.7 Navigation and Independent Pages System

In this application, the decision was made to implement **independent pages** with no direct links between them. This approach was chosen for several reasons related to code structure and user experience.

- **Easier Code Structure:** Dividing the application into independent pages made the codebase more manageable. Each page handles a specific functionality, simplifying development, debugging, and adding new features without affecting other parts of the app.

- **Avoiding Infinite Scroll:** Long, infinite scrolling pages can be difficult to navigate, especially when dealing with complex data and multiple visualizations. By breaking the content into separate pages, the interface remains clear and focused on one task at a time.
- **Clear Separation of Tools:** Each page is dedicated to a specific tool (e.g., similarity search, embeddings analysis with clustering and dimensionality reduction,...). This separation helps users focus on one feature at a time and reduces confusion.
- **Intuitive Navigation:** All navigation links are contained within their respective pages, making the interface more intuitive. Users can easily switch between different features without needing to return to a central homepage.

0.8 User Interface

The interface enables **interactive exploration** of clustering and dimensionality reduction results. Users can zoom into clusters, adjust parameters, and filter the dimensionality space to gain insights.

Optimizing User Experience

To enhance the user experience:

- Pre-calculated figures and variables minimize waiting times when navigating.
- Some elements, like similarity matrices, cannot be stored due to high memory demands.
- Titles and subtitles clarify each section's purpose.
- Loading circles provide feedback when processes are ongoing.

Design Choices

Clustering and dimensionality reduction are presented on the same page to:

- Minimize space usage.
- Efficiently apply algorithms with compact controls.

Compact and Efficient Interface

The interface supports:

- Five dimensionality reduction algorithms.
- Three clustering algorithms.
- Two datasets (embeddings and facial features).

All functions are integrated with only two plots for clarity and efficiency.

0.9 Clustering Algorithms

The tool supports three clustering algorithms: **K-Means**, **DBSCAN** (Density-based Spatial Clustering of Applications with Noise), and **Agglomerative Hierarchical Clustering**. These algorithms differ in how they define and form clusters, offering various perspectives on how to group the data.

- `K-Means` is a partition-based method that requires specifying the number of clusters in advance. It works well for spherical clusters.
- `DBSCAN` is a density-based algorithm that does not require a predefined number of clusters. It can identify clusters of arbitrary shape and handle noise.
- `Agglomerative Hierarchical Clustering` is a bottom-up approach that builds a hierarchy of clusters. The user can decide the number of clusters based on a dendrogram.

Each of these algorithms brings unique strengths to the clustering process, making the tool versatile. `K-Means` excels in scenarios where clusters are well-separated and roughly spherical, offering a fast and simple solution. `DBSCAN` is particularly effective for datasets with irregularly shaped clusters or noise, as it does not assume a specific cluster shape. `Agglomerative Hierarchical Clustering` provides a hierarchical view of the data, enabling users to explore relationships between clusters at different levels of granularity.
Together, these algorithms cover a wide range of clustering scenarios, ensuring flexibility and adaptability to different dataset characteristics. This comprehensive selection allows users to extract meaningful insights from diverse types of data structures.

0.10 Dimensionality Reduction Algorithms

I selected five dimensionality reduction algorithms for the tool: **LLE** (Locally Linear Embedding), **MDS** (Multidimensional Scaling) – both metric and non-metric versions, **PCA** (Principal Component Analysis), **t-SNE** (t-Distributed Stochastic Neighbor Embedding), and **ISOMAP** (Isometric Mapping).
These algorithms cover a range of linear, non-linear, global, and local techniques, offering diverse perspectives for analyzing the high-dimensional embedding space.

- `PCA` is a linear method that projects data into the principal components, making it useful for capturing the most significant global variance in the data.
- `LLE` and `ISOMAP` are non-linear methods that aim to preserve the local structure of the data, making them useful for complex, non-linear relationships within the data.
- `MDS` can be used in both its metric and non-metric forms to represent similarities or distances between data points in a lower-dimensional space. Metric MDS preserves the exact distances, while non-metric MDS focuses on preserving the rank order of distances.
- `t-SNE` is another non-linear technique that focuses on preserving local structures while mapping the data to a 2D or 3D space, making it particularly effective for visualizing complex datasets.

This range of algorithms allows for a comprehensive analysis of the dataset, each offering different advantages in terms of preserving global or local data structures. By applying multiple techniques, users can gain a deeper understanding of the latent space and explore different relationships and patterns within the embeddings.

Chapter 3 - Results for Dimensionality Reduction Algorithms & Clustering

0.11 Analysis of Clusters in the Embedding Space

For this analysis, I filtered the dataset to include only bald individuals, reducing the dataset from 30,012 images to 423. This subset is more manageable for computationally intensive algorithms. I determined that approximately 19 clusters should be used, as each celebrity has more than 20 images. I do this because I expect to cluster celebrities in this case.

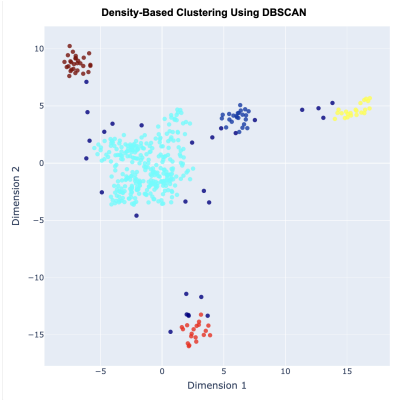

Figure 6: DBSCAN Algorithm

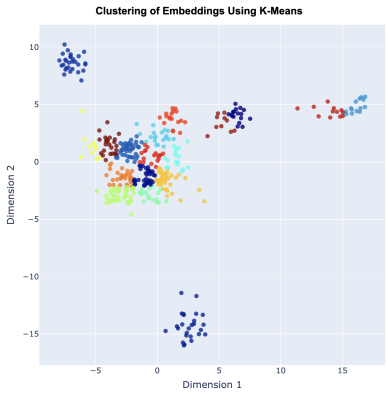

Figure 7: K-Means Algorithm

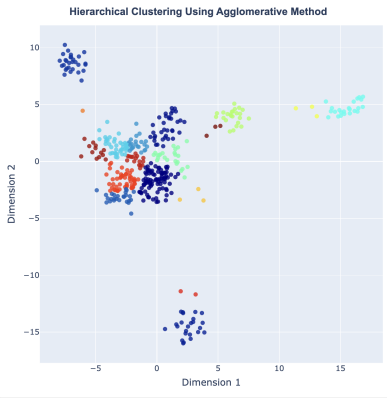

Figure 8: Agglomerative Algorithm

Figure 9: Clustering Algorithms Applied to PCA Embeddings

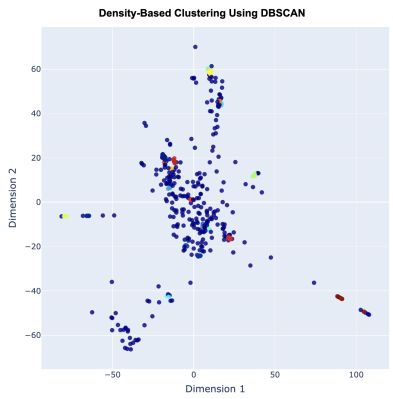

Figure 10: DBSCAN Algorithm

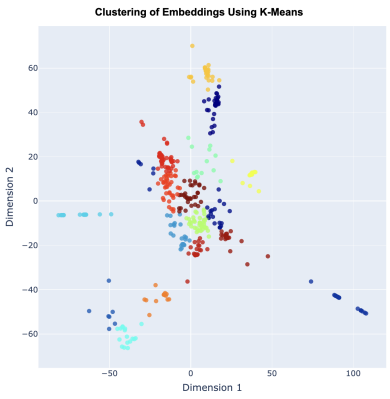

Figure 11: K-Means Algorithm

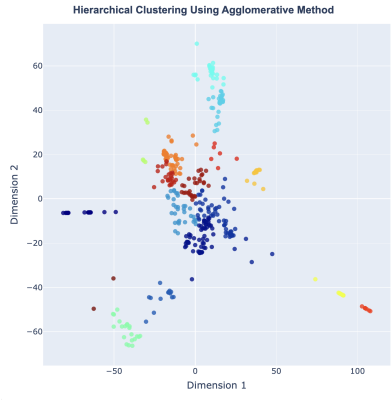

Figure 12: Agglomerative Algorithm

Figure 13: Clustering Algorithms Applied to ISOMAP Embeddings

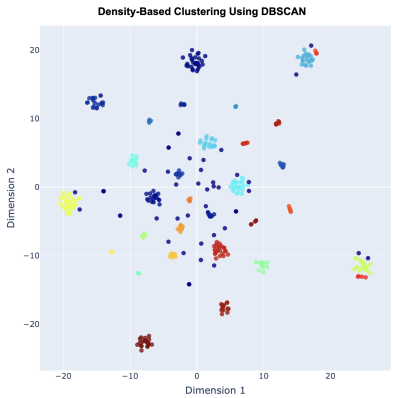

Figure 14: DBSCAN Algorithm

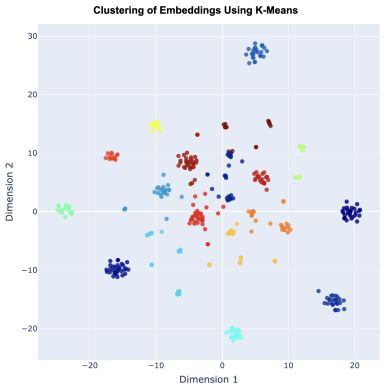

Figure 15: K-Means Algorithm

Figure 16: Agglomerative Algorithm

Figure 17: Clustering Algorithms Applied to t-SNE Embeddings

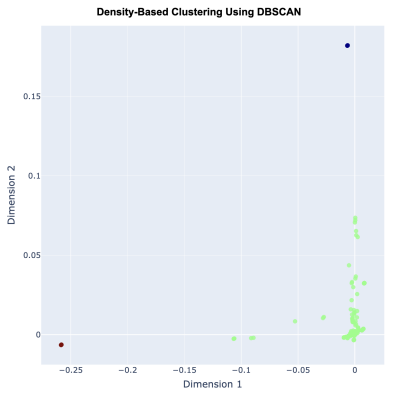

Figure 18: DBSCAN Algorithm

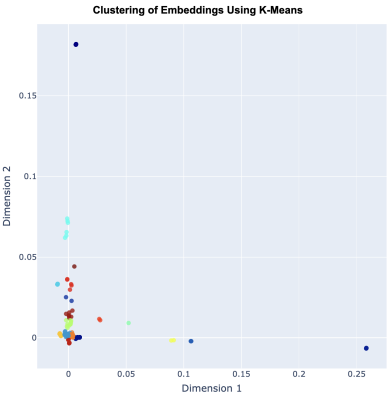

Figure 19: K-Means Algorithm

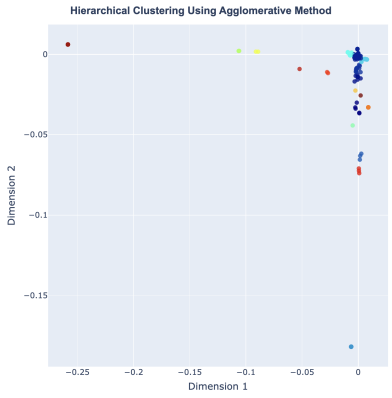

Figure 20: Agglomerative Algorithm

Figure 21: Clustering Algorithms Applied to LLE Embeddings

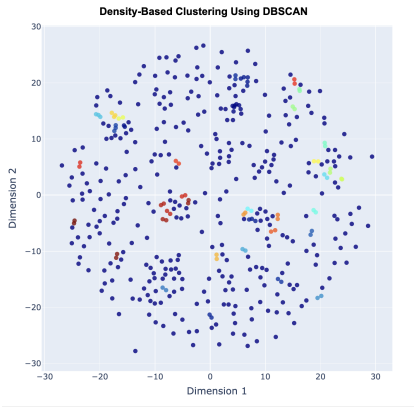

Figure 22: DBSCAN Algorithm

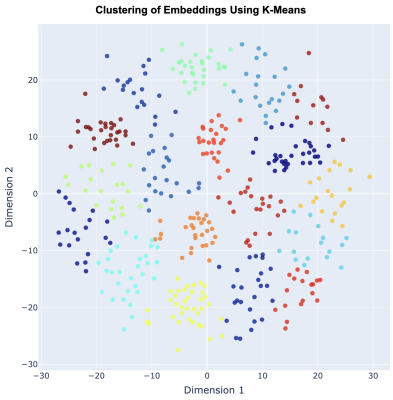

Figure 23: K-Means Algorithm

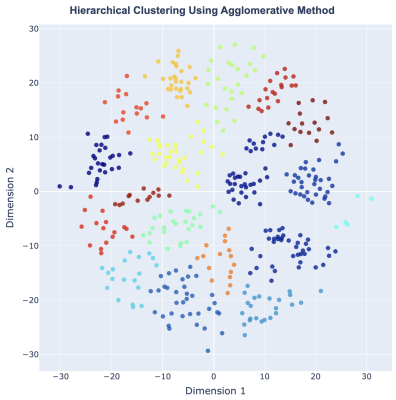

Figure 24: Agglomerative Algorithm

Figure 25: Clustering Algorithms Applied to MDS Embeddings

As seen in all these plots, the clustering strongly depends on the dimensionality reduction technique used, and thus the reduced embedding space, since the dimensionality reduction is applied first here. The two best-performing algorithms are t-SNE and MDS, which, in their own way, manage to isolate the celebrities. MDS forms an oval shape where everything seems mixed, while t-SNE creates distinct small isolated clusters. In terms of clustering algorithms, the one that consistently performs the best is hierarchical clustering, followed closely by K-Means, which performs quite well for most cases.

0.12 Analysis of Clusters in the Facial Features Space

For the PCA plot of the filtered features (baldness and blond hair), clear separation between the two groups is observed. The lower-right region is dominated by individuals with blond hair, while a diagonal movement toward the upper-left reveals a gradual transition to bald individuals. In this case, we aim to identify two distinct clusters: one for bald individuals and the other for those with blond hair. Figure 29 shows the clustering results using the three algorithms with their best-tuned parameters for this embedding. The Agglomerative clustering with complete linkage performs exceptionally well, effectively distinguishing the two groups. On the other hand, K-Means struggles due to its assumption of spherical clusters, leading to numerous misclassifications. DBSCAN performs poorly here, as the embedding space's density is not suitable for this density-based method.

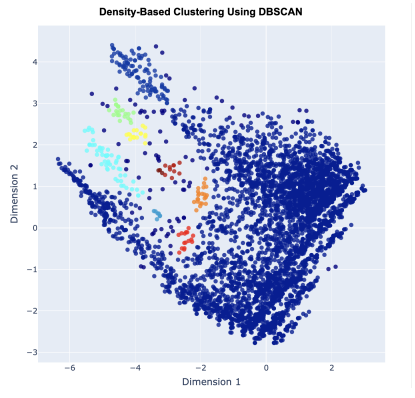

Figure 26: DBSCAN Algorithm

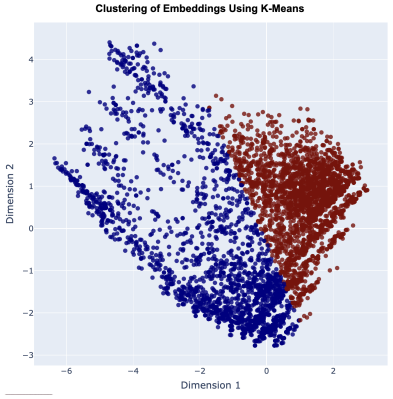

Figure 27: K-Means Algorithm

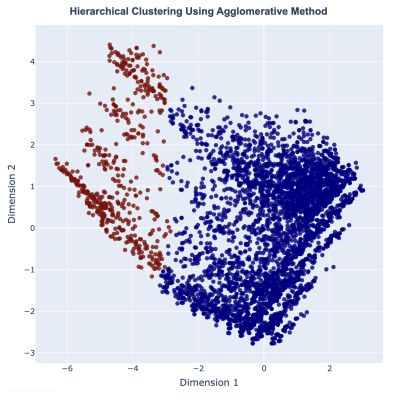

Figure 28: Agglomerative Algorithm

Figure 29: Clustering Algorithms Applied to PCA Embeddings

Chapter 4 - Ideas for Future Improvements

- Implement a dendrogram visualization for hierarchical clustering to allow users to visually inspect the relationships between clusters. This would help in selecting the optimal number of clusters, especially for large datasets like CelebA.

- Introduce clustering evaluation metrics such as the Silhouette Score to allow users to assess the quality of clustering results directly within the interface. This would help in fine-tuning clustering parameters.
- Implement real-time visualization during the clustering process, showing how the algorithm evolves, which enhances the user experience and provides immediate feedback, especially with large datasets.
- Introduce vector arithmetic for embeddings to explore latent space manipulations. For example, subtracting the embedding of "glasses" and adding the embedding of "smile" to an image of a man without a smile but wearing glasses, and identifying the closest match, would allow users to verify if the result is a man smiling without glasses. This would provide an intuitive way to understand and experiment with embedding representations.
- Allow users to select specific clusters and derive statistical insights, such as average feature values, standard deviations, or attribute distributions. This feature would help in understanding the characteristics of each cluster and drawing meaningful conclusions about the dataset.
- Allow users to perform clustering before dimensionality reduction and then apply the dimensionality reduction. This would enable users to test how well clustering results generalize to different reduced spaces, offering greater flexibility in analyzing how data structure changes across different spaces.
- Enable the export of results and visualizations in multiple formats, including CSV for filtered embeddings, JSON for cluster assignments, and PDF for plots, facilitating further analysis and reporting.

Chapter 5 - Citation of Toolboxes and Sources

The following toolboxes, sources, and papers were used in developing the software:

Libraries:

- **Scikit-learn:** Used for implementing dimensionality reduction algorithms (PCA, t-SNE, etc.) and clustering algorithms (K-Means, DBSCAN,...).
- **Plotly:** Used for creating interactive plots and visualizing data.
- **Pandas and NumPy:** For data manipulation and numerical operations.
- **Dash:** Used for building interactive web applications, integrating the various algorithms and visualizations.
- **Dash-Bootstrap-Components:** Provides Bootstrap components for creating a clean and responsive layout for the web application.

Sources and Papers:

- Oshima, T., Takagi, K., & Nakata, M. (2024). *Clustering-friendly Representation Learning for Enhancing Salient Features.*
- Dash Documentation & User Guide | Plotly
- OpenAI's *ChatGPT*: Used for generating code snippets.
- Lecture slides on Dimensionality Reduction from the course *LDATA2010 - Information Visualisation.*
- Lecture slides on Clustering from the course *LDATA2010 - Information Visualisation.*

Chapter 6 - Appendix

All the images presented below are taken from the application developed for embedding analysis and clustering. They illustrate the various pages and features available to the user.

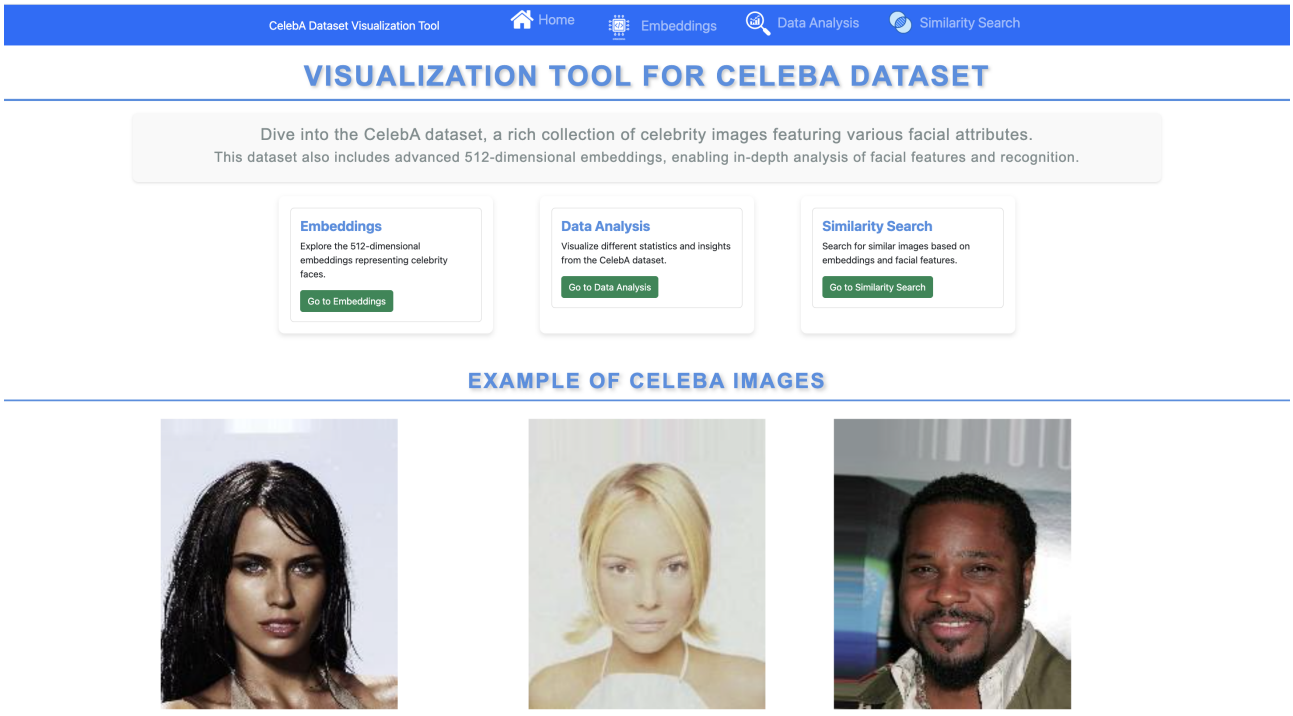

Figure 30: Home Page

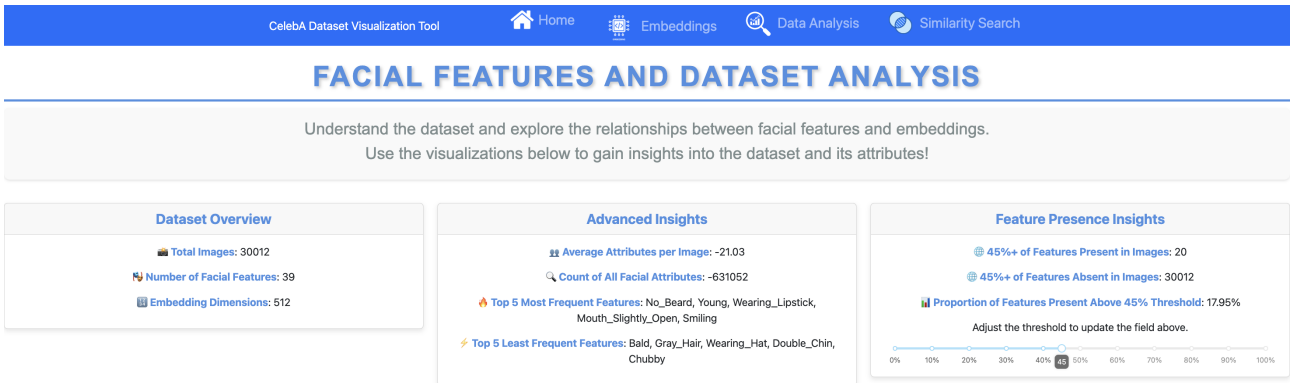

Figure 31: Facial Features Analysis Page - Overview

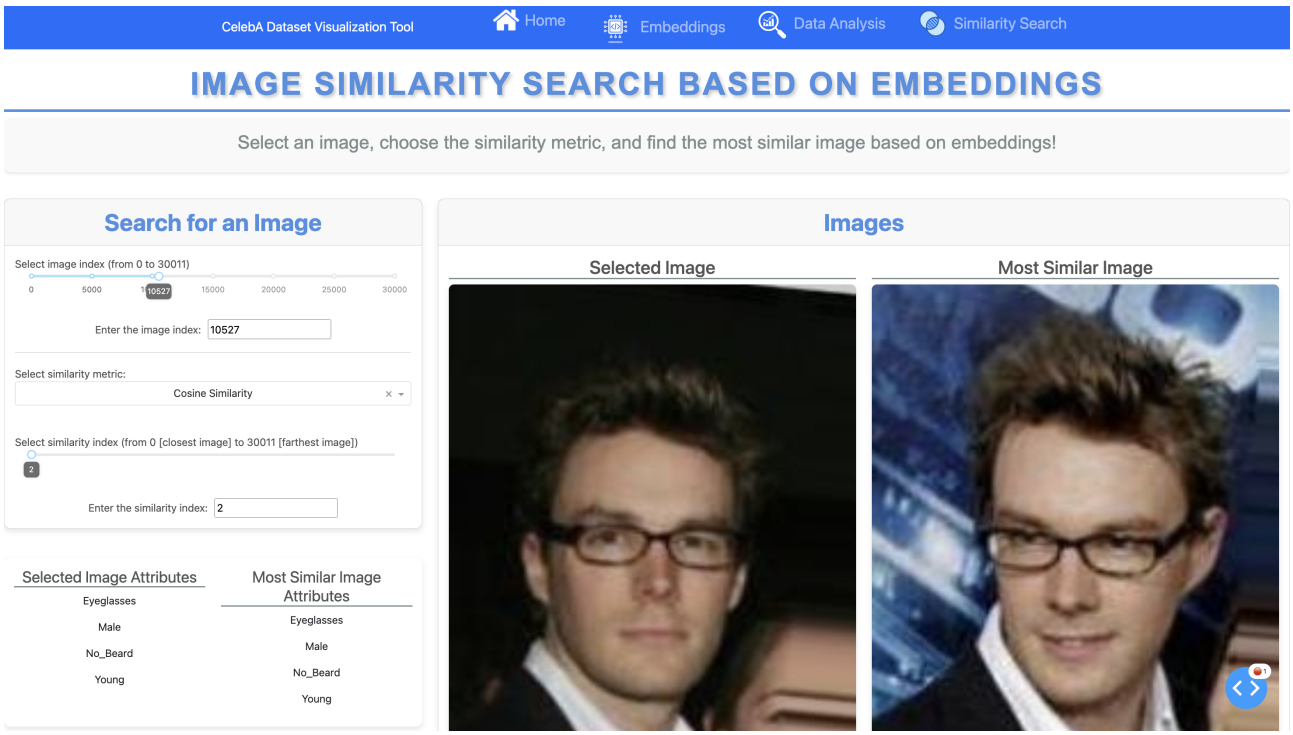

Figure 32: Similar Search Page

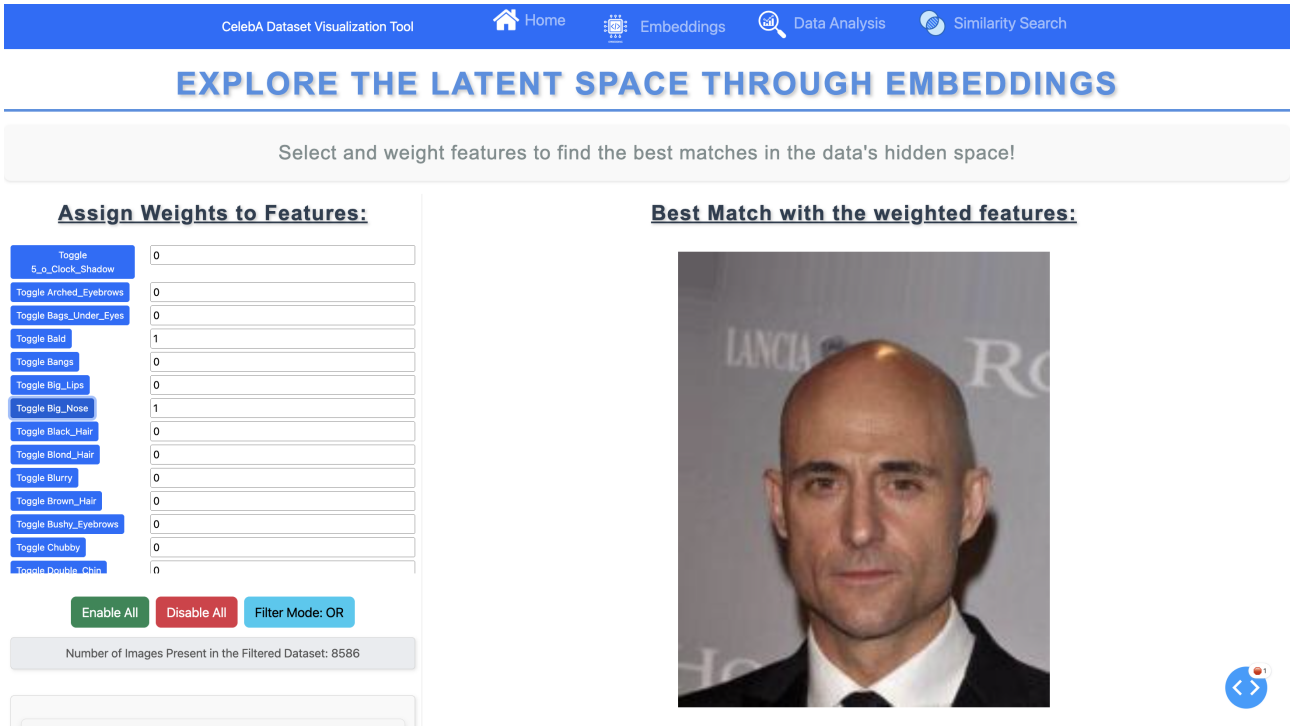

Figure 33: Embeddings Page - Exploring Latent Space

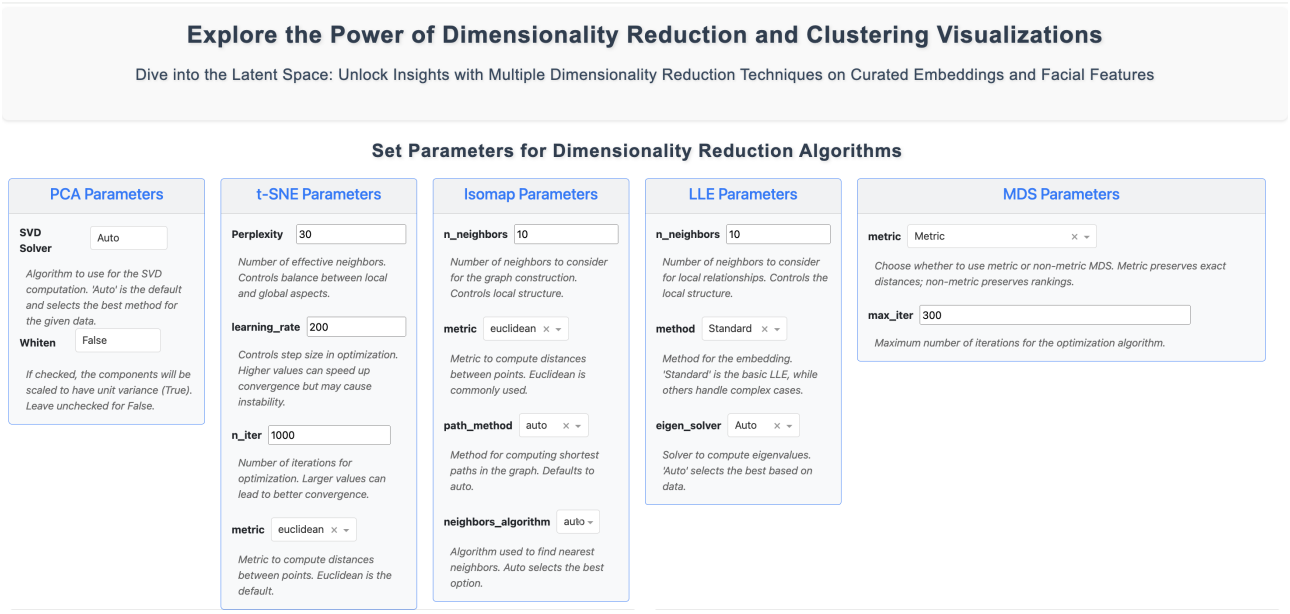

Figure 34: Embeddings Page - Algorithm Parameters

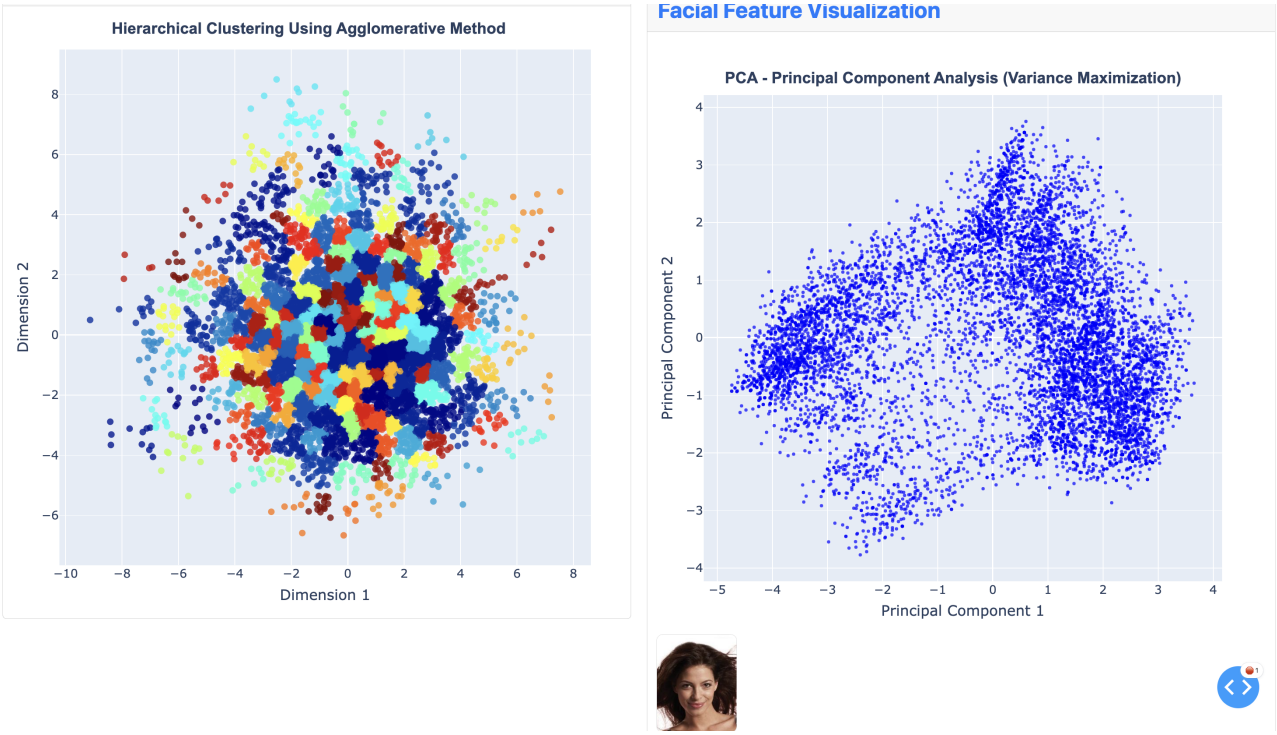

Figure 35: Embeddings Page - Dimensionality Reduction & Clustering