LDATA2020:
CelebA-Vis: User Guide

Farah Aroud

December 2023

1 Introduction

Have you ever wanted to build your own face recognition software? Do you have access to the CelebA dataset but do not understand how the embeddings describe each image? Look no further! CelebA-Vis is a visualisation software designed specifically to explore that dataset.

2 Requirements

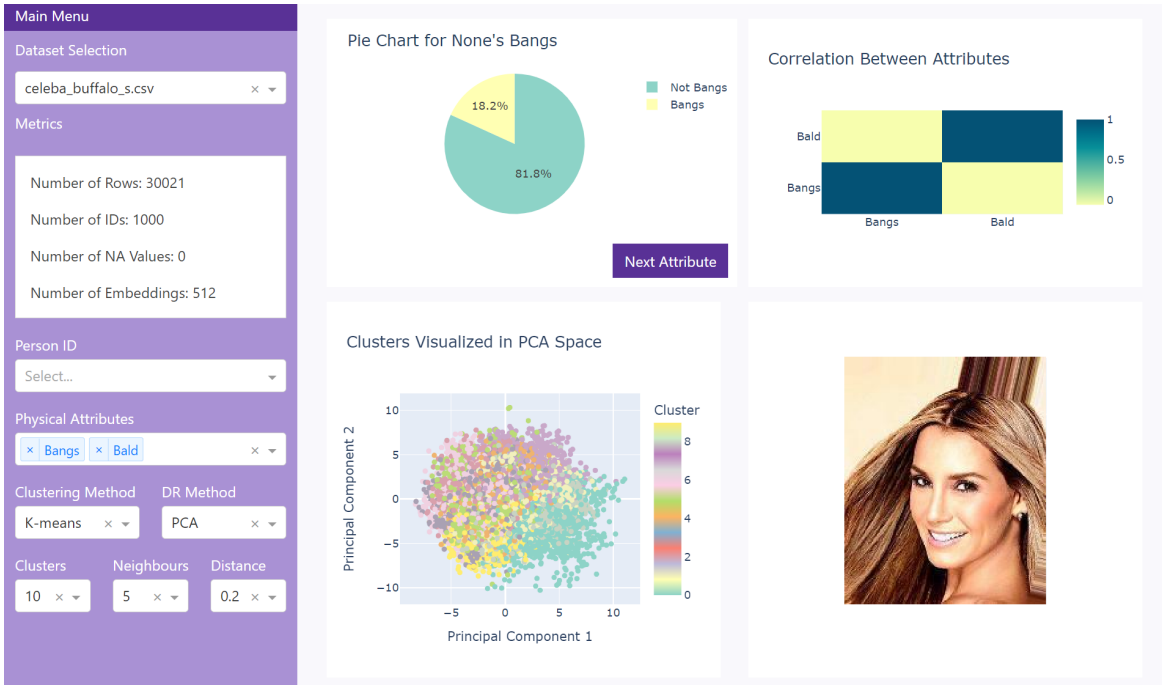
Before using our software, you must make sure that you have installed Python on your machine. You can then download the CelebA dataset, available online. Once that is done, you can install the needed libraries by using the `requirements.txt` file. It was generated using the `pipreqs` command. Finally, you must make sure that the downloaded files have the following structure:

```
Project
├── app.py
├── requirements.txt
└── celeba
```

Now we are all set to start exploring the data. To start the software, if the terminal change the directory to go to the `Project` folder then, run `python app.py`.

Future Update: You might have noticed that the visualisation tool takes a few seconds to launch, you can expect a faster launch time in a future version.

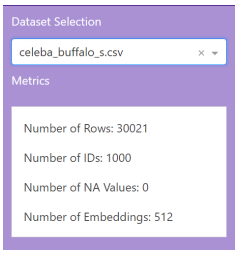
3 Software Layout



There are 2 parts to the website: the Main Menu, on the left, and the Plots, on the right. The Main Menu allows you to choose different parameters to update the visualisations on the right. This layout was generated by following the tutorial in [2]. ChatGPT was also used to assist in placing elements in the layout.
Future Update: You might notice that the layout changes a little bit on different screen sizes, you can expect a more adaptive layout in a future version.

4 Basic Exploratory Features

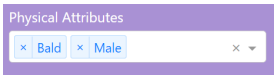
4.1 Choosing the Dataset



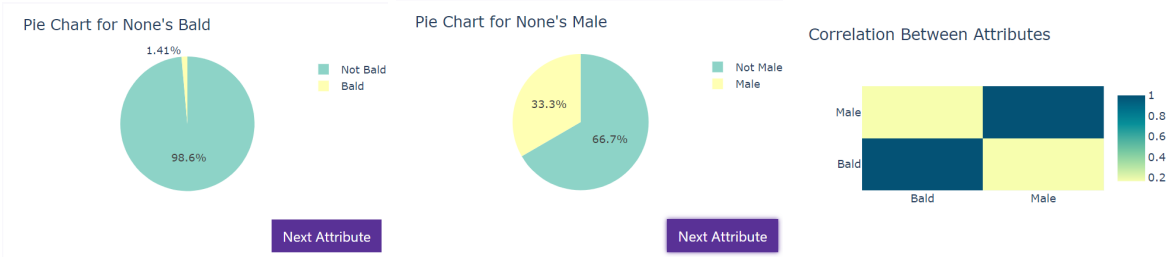
Now that you have an overview of the software layout, we can jump into the dataset exploration. The CelebA database is composed of 2 datasets: `celeba_buffalo_s.csv` and `celeba_buffalo_l.csv`. The first question that you may ask yourself is "What makes those two datasets different?". The software leaves the possibility to choose the dataset that you want to study with the drop-down "Dataset Selection" on the left. Once the dataset is selected, a few metrics are computed and shown in the box under. The plots are also reloaded to adapt to the chosen dataset.

To make sure that the system responds quickly, these values are precomputed.

4.2 General Exploration



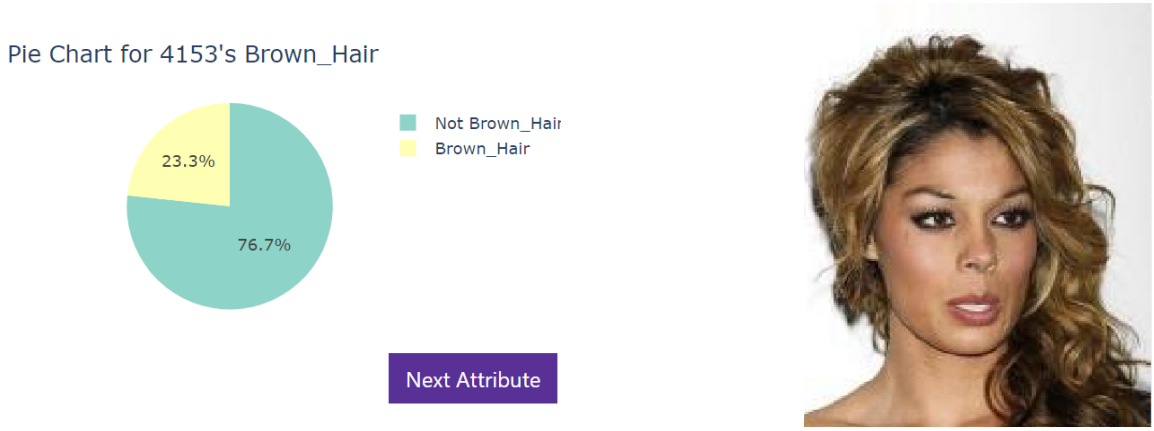
Once the dataset is chosen, you can start exploring the features space. In the two datasets, there are some binary features that describe the physical attributes of the person on the picture. One may want to know what proportion of person in the dataset is wearing glasses, or how is being bald related to being a man. The attributes you want to explore can be chosen under "Physical Attributes" in the Main Menu. Doing this will update the pie chart and correlation matrix. When you choose multiple attributes, you should click on the "Next Attribute" button to view each pie chart. Here is an example of the charts generated by selecting the attributes "Bald" and "Male".



Future Update: You might notice that the titles of the pie charts include the word "None". That is because they are computed on the whole dataset but can also be plotted for a selected id. You can expect more adapted titles in a future version.

4.3 Exploration by ID

Once you have gotten an overview of the dataset, you can also explore the proportions of attributes for a chosen ID. To do that, you can choose an ID in the Main Menu under "Person ID". This will update the pie charts and the picture in the right corner. This functionality is useful to understand that a same person can sometimes have a different hair colour, or can sometimes be frowning and sometimes be smiling. Selecting ID 4153 and the Brown_hair attribute yields the following changes:



Future Update: As of today, the only picture you can view for a chosen ID is the first appearing picture in the dataset. You can expect to be able to choose between all pictures of a selected Id to view in a future version.

5 Clustering and Dimensionality Reduction

When you are done with the general exploration, you have a better knowledge of the composition of the dataset. But what are the embeddings used for? Well, in the context of facial recognition, we want to "cluster" the images with faces that look alike. As we have seen in the previous sections, some of a persons' physical attributes can change over a period of time. Therefor, it is not enough to analyse these binary features to try and find similar looking persons.

The CelebA-Vis software lets you choose between three different clustering algorithms to help you make more sense of the use of embeddings. The following subsections are meant to help you understand these algorithms and show you outputs that they can produce.

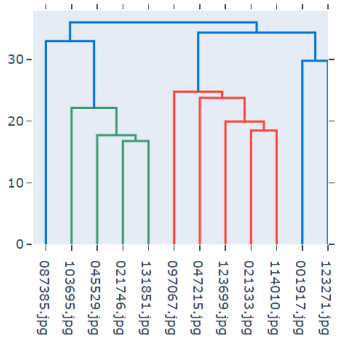
5.1 Hierarchical Clustering

This first algorithm was implemented using `plotly.figure_factory`'s `create_dendrogram()` function. This function computes the hierarchical clustering and outputs a dendrogram. A dendrogram is a tree shaped graph. The values on the y axis correspond to the distance between each cluster.

To use the Hierarchical Clustering in the CelebA-Vis software, you must select at least 3 attributes to filter the dataset first. Then, you can choose "Hierarchical" under "Clustering Method" in the Main Menu. That will apply the clustering to a sub dataset of images having each chosen attribute set to 1.

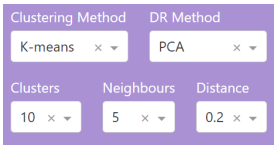
Future Update: This implementation is the least successfully implemented one in the CelebA-Vis software. In a future version, you may expect the following changes. First, when choosing the attributes, one should be able to also choose their values. I wanted to implement that selection by clicking on each pie chart to select the value but was unsuccessful. Next, this graph lacks interactivity, I was unable to make sure that by clicking on a branch representing an image, the corresponding image would display next to the plot. Finally, it would also be nice to modify the plot colours to match the rest of the layout.

Here is an example of the produced dendrogram using attributes Bald, Goatee and Eyeglasses:

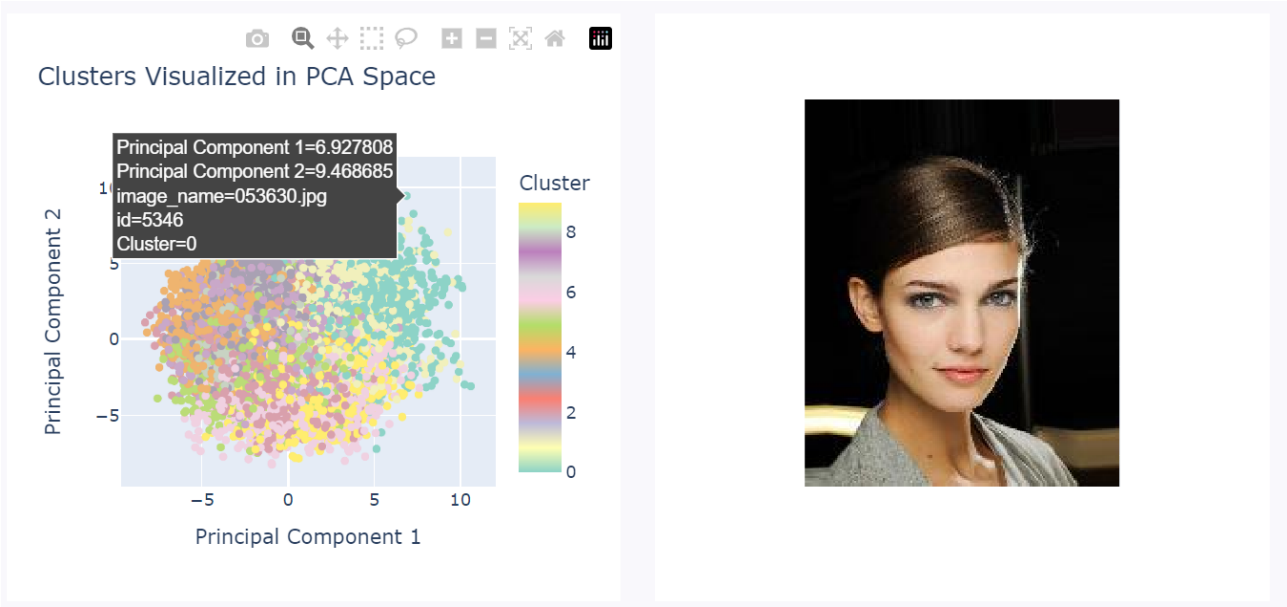


5.2 K-means

This next algorithm was implemented using `sklearn.cluster`'s `Kmeans`. K-means algorithm starts by selecting N centroïds. It then clusters the data points by least distance and recomputes the centroïds. It is an iterative method and this step is repeated until convergence is reached. Since this algorithm is very demanding in terms of computations, it can take quite some time to run. To improve on that, it is possible to use dimensionality reduction (DR) algorithms. In this software, you can choose between PCA or UMAP to perform that task. The number of components generated by PCA is set and was calibrated for each dataset to keep 95% of explained variance. It is therefor essential to apply another PCA after the clustering algorithm to be able to plot the data. On the other hand, UMAP will always reduce to 2 features. UMAP is a non-linear DR algorithm. It is faster than t-SNE [3] and allows to reduce data while keeping the same local density in clusters.



To use the K-means functionality, you must first set the Clustering Method to K-means and choose the number of clusters. You can then choose the DR Method. Once this is done, the clusters will be computed and shown on the bottom left plot. Here is an example of output with K-means and PCA.

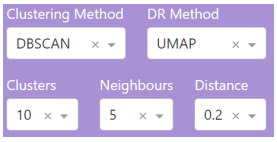


When the plot appears, you can interact with it. Hovering your cursor over a dot will give you information such as the persons' id or the name of the image file. You can also click on the dot to display the corresponding image on the right.

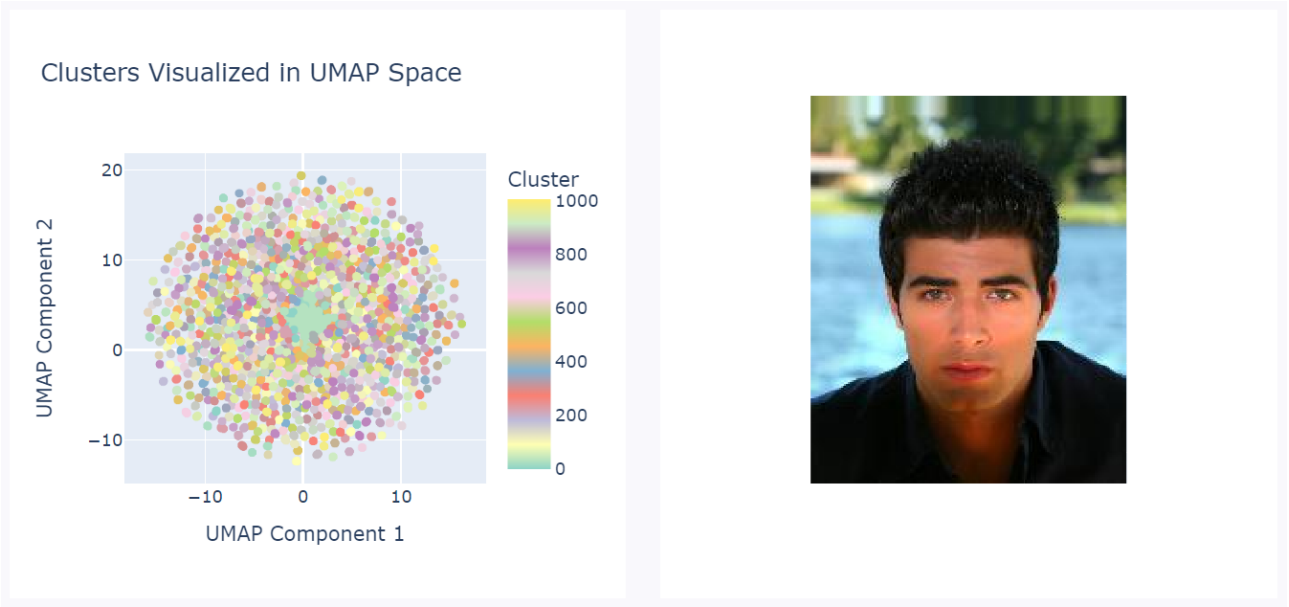
Future Update: To be able to use the click-to-display-image functionality, no person ID should be selected in the Main Menu. You can expect to be able to use that feature without this inconvenience in a future version.

5.3 DBSCAN

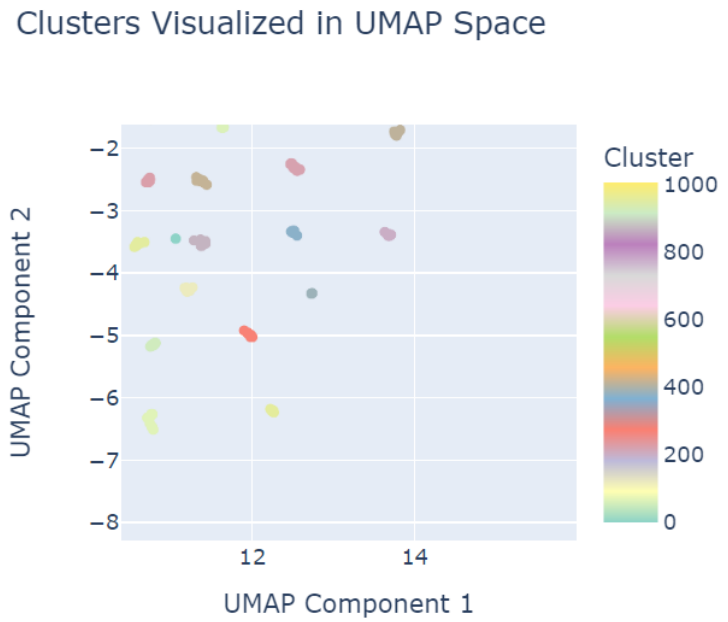
This last algorithm is a density-based clustering algorithm, "it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away)" [1]. It was implemented using `sklearn.cluster`'s `DBSCAN`. Just like K-means, you can choose a DR algorithm to apply before the clustering algorithm.



To use the DBSCAN functionality, you must first set the Clustering Method to DBSCAN, choose the number of neighbours to consider and the maximum distance between two samples for one to be considered as in the neighbourhood of the other. You can then choose the DR Method. Once this is done, the clusters will be computed and shown on the bottom left plot. Here is an example of output with DBSCAN and UMAP.



The DBSCAN plot has the same interactive principles as the K-means plot. When the plot appears, you can interact with it. Hovering your cursor over a dot will give you information such as the persons' id or the name of the image file. You can also click on the dot to display the corresponding image on the right. You can also zoom in on the plot. This feature is quite interesting in the case of DBSCAN as the algorithm can produce a lot of small clusters. Here is a zoomed-in view of the preceding plot.



5.4 The Authors' Preference

You must keep in mind while exploring that, while you can mix and match clustering algorithms and dimensionality reduction, some combinations are more efficient. I have found through experimenting that using DBSCAN with UMAP dimensionality reduction gives the best clustering results.

6 Conclusion

With the usage of data becoming more prevalent every day, we must find ways to visualise data and understand it. Indeed, the values of a latent space generated by a Deep Neural Network, do not make sense to the human eye, but they can be valuable for different tasks such as clustering or classification.

With CelebA-Vis, you can superbly visualise the CelebA database and understand how facial recognition works. You can immerse yourself in clustering tasks and compare their outputs to find the one most suited to your desired outcome. Although it is not flawless, it is a nice tool to satisfy a curious user. As for any software, some updates can be expected to improve the reaction speed, particularly when choosing a new dataset or a new clustering method. I also expect to make the hierarchical clustering more interactive.

References

[1] *DBSCAN*. `https://en.wikipedia.org/wiki/DBSCAN`. Accessed: 2023-12-18.

[2] *How to Create a Dashboard with Dash and Plotly*. `https://sakizo-blog.com/en/487/`. Accessed: 2023-12-18.

[3] *t-SNE and UMAP projections in Python*. `https://plotly.com/python/t-sne-and-umap-projections/`. Accessed: 2023-12-18.