Advanced Visual Data Analysis
Lab Assignment 3

Anna Granberg & Filip Hamrelius

anngr950 & filha243

1. Image comparison

1.1. Method

The feature vectors to compare the images were constructed of the following features:

- Mean RGB value across entire image
- Mean RGB value around the middle point (20 x 20 pixels)
- Mean RGB around the middle point of each quadrant (20 x 20 pixels)
- Luminance across image
- Edge detection (Density, Mean, Intensity and Orientation)

This was implemented in Python with the libraries CV2, NP and PIL. To calculate the similarity Cosine similarity was used, as a quality check the euclidean distance was also calculated. Each feature was calculated in a loop that went through each image, the data were saved in an array. All data were then concatenated to a large matrix containing each feature vector for each image. This was later used to calculate the distance and create the distance matrix.

1.2. Results

Figure 1 - Cosine similarity

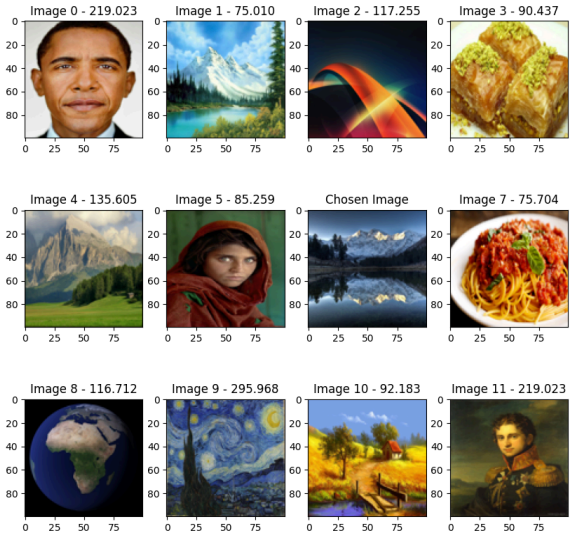

Figure 2 - Euclidean distance

1.3. Discussion

As mentioned above, the main method utilized was cosine similarity. This method has proven to be efficient in the past, as it compares each feature in a multidimensional space by calculating the cosine of the angle between two vectors. This calculation was done between each feature, a mean of this was then calculated and is what is shown in *Figure 1* and this in turn laid the ground to construct the distance matrix. For good measure a euclidean distance function was also implemented, mostly to compare the results, see *Figure 1 & 2*. No weights were used for the result shown in the figures. Depending on the dataset it can be highly relevant to choose what features are more important. For example if only classifying landscapes, colors might be similar over all and edges can prove a better feature to differentiate between the images.

2. Text analysis

1.1. Method

To extract, process and visualize the data, Python was used. This included using libraries like: Pandas, nltk, sklearn and matplotlib. The topic model used was a LDA (Latent Dirichlet Allocation) provided by sklearn.

1.1.2 Temporal Data

The implementation followed the instructions closely and the pipeline looked like this:

- Clean the data, tokenize it, and remove stopwords.
- Read a list of keywords from a text file and filter the preprocessed data to include only rows containing these keywords.
- Count the occurrences of each date in both the original and filtered data.
- Plot the distribution of content over time for both the original and filtered data using bar plots.

The words used to filter data was created by Chat-GPT-3.5 where it was given the example words and asked to create a list of 350 similar words. These were then used to create the subset.

1.1.3 LDA

The implementation followed the instructions closely and the pipeline looked like this:

- Clean the data, tokenize it, and remove stopwords.
- Use TfidfVectorizer to transform the preprocessed text into a TF-IDF matrix.
- Instantiate a LDA to perform topic modeling.
- Use the LDA model to assign topics to each document.
- Group Documents by Topic and Date: Group the documents by their assigned topics and dates.

1.2. Results

```
Content: dread threat
Content: explosive explosive explosive explosive explosive explosion explosive disaster explosion explosion explosion
Content: threat terrorism fear terrorist security security terrorism catastrophe security security security security security terrorism security fear terrorism
Content: emergency violence fear
Content: security dread attack
Content: attack explosive military explosion explosion disaster
Content: military security
Content: security threat vulnerability danger fear dangerous
Content: threat threat security
Content: catastrophe security terrorist attack security explosion security targeting security security vulnerability security explosion terrorist intelligence terrorism security
Content: attack
Content: surveillance vandalism
Content: explosion
Content: dead dead
Content: emergency emergency terrorist terrorist
Content: security
Content: explosion bombing destruction terrorism terrorist target explosion
Content: dangerous
Content: terrorism terrorism terrorism terrorism threat terrorism intelligence counterterrorism terrorist catastrophe terrorist
Content: terrorism bombing terrorism disaster explosion terrorist explosion terrorist terrorism victims terrorism security security security bombing attack military terrorism explo
y security security
Content: security security dangerous
Content: security security chaos intervention radical
Content: conspiracy
Content: dangerous military security
Content: radical military
Content: explosion terrorist explosion terrorism
Content: attack target targeting
Content: threat dangerous
Content: dread target chaos dread military terrorist attack
Content: explosive security
Content: terrorist
Content: chaos
Content: security threat threat attack dangerous
Content: security security
Content: violations
Content: carnage
Content: terrorism bombing terrorism disaster explosion terrorist explosion terrorist terrorism victims terrorism security security security bombing attack military terrorism explo
y security security
Content: explosive explosive explosive explosive explosion explosive disaster explosion explosion explosion
Content: terrorism victims chaos emergency
Content: threat
Content: dangerous explosive explosive explosive
Content: military
```

Figure 1 - Filtered Data represented by content feature

Figure 2 - Temporal data: Data vs Filtered data

```
Topic 1:
fire, stanley, robbery, computers, threats
Topic 2:
games, city, festival, vastopolis, thom
Topic 3:
mayor, emails, fish, money, group
Topic 4:
flu, teens, systems, quick, commission
Topic 5:
said, crash, investigators, police, flight
```

Figure 3 - Top 5 words by topic

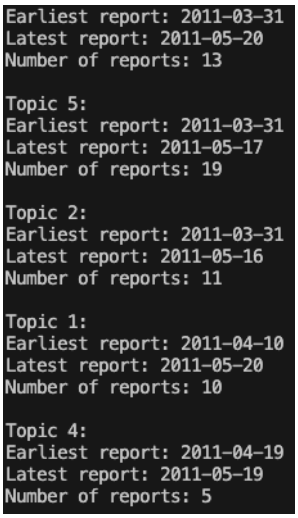

Figure 4 - Topics in relation to temporal data

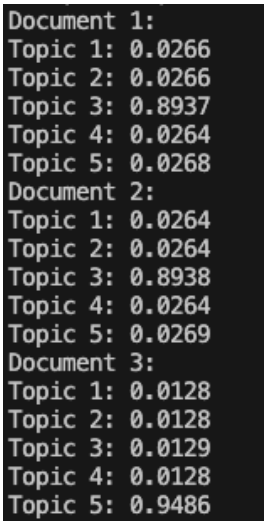

Figure 5 - Topics in relation to documents

1.3. Discussion

The order of topics has nothing to do with the most informative one. It depends on the objective of the research and how the data is formatted. It is worth mentioning that the topics used here are in early stages and are still quite broad, later topics might be more nuanced than the first ones.

By comparing the filtered data in respect to the temporal data in figure 2 compared to the LDA created topics in regard to temporal data in *figure 4* it can be seen that the topics range in a really broad timespan. For example, the topic 4 which includes words like “police, explosive, security, bomb” has its earliest report 2011-04-10 and latest 2011-05-20. The filtered data do have peaks for these two reports specifically, although it is a lot easier to read from the filtered data. The problem with LDA is that each document has a date connected to it, and that document is then connected to a topic, see *Figure 5*. But each topic does not have a specific date, only a timespan of the documents. The first method is therefore easier to comprehend, it also allows the user to filter on specific keywords rather than creating bins for general recurring subjects, see *Figure 3*. It is worth mentioning that LDA would prove more efficient if we did not know what we were looking for, identifying topics not known is a really big advantage.