Content-based analysis

Hassan Mustafa,
Hasmu396@student.liu.se
TNM098

1. Background

This report presents an analysis of two content-based analysis tasks involving text and image datasets. The first task is to compare an image to all others in a dataset and rank them based on similarity. The second task requires identifying plagiarized text and listing the source and destination for each copied element.

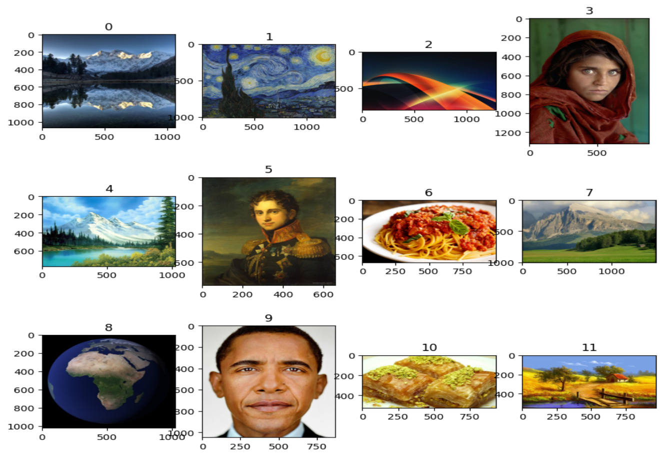
1.1 Data

The data for these tasks consists of 12 .jpg files for the image analysis task and 10 .txt files for the plagiarism detection task.

2. Approach

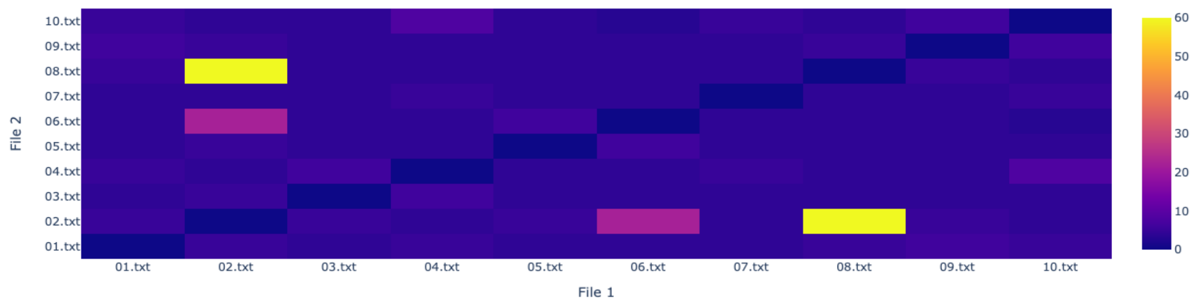
2.1 Task 1

To compare and rank images based on their features, we extract relevant information from the images. We select the color content, luminance distribution, and edges of the images as the features for comparison. We extract the color content of an RGB image by computing a 3D histogram and flattening the result to a one-dimensional array. Using the YUV color space, we extract the luminance information. To extract the edges of the image, we convert it from RGB to grayscale and apply a Gaussian blur filter, which allows us to detect edges using an edge detection algorithm. We then use the Euclidean distance to compare feature vectors and rank the images based on their closest values. We also add weighting based on the importance of the features to give some bias to the feature we deem to be more important. Below you can see the output for the image indexed as 0.

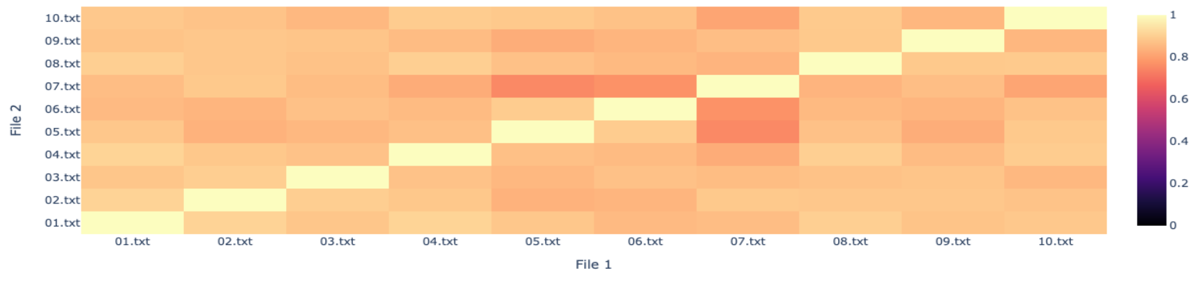


2.2 Task 2

In order to identify plagiarized content, the text is preprocessed by eliminating punctuation, converting it to lowercase, and transforming it into a single-line string. A longest common substring algorithm is used to discover shared substrings among the text files. Subsequently, a heatmap is employed to depict the extent of the longest common substrings between each text file, as depicted in the figure below.



Even though the task description didn't explicitly request it, I decided to include cosine similarity as a measure to ensure we capture similarities in writing style. This was to prevent missing instances where the same idea is expressed using different vocabulary. The figure below illustrates this approach.



Here we see that text file 7 has some similarity with some of the other text files, which we could not see in the previous method.

3. Discussion

In Task 1, the difficulty lies in selecting which features to compare and how to optimally assign weights to them. To address this challenge, a possible solution is to employ a machine learning algorithm that can optimize a similarity metric for the highest possible similarity score. In contrast, for Task 2, the focus is on optimizing computation time. Since we are only comparing 10 text files, this may not be a challenge for our case, but it can become more difficult when dealing with larger datasets and some measures of optimizing the solution should be taken into consideration.