Candidate: Teodor Barbuceanu **Date: 16.04.2025**

Task: Task 1 – Logo Similarity

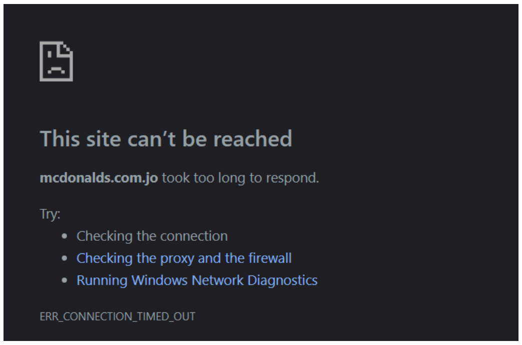
0. Why did I choose this task?

I had some previous experience with photo analysis of objects, but the main reason was that I wanted to also ask myself, after designing a lot of logos, what makes them stand out? So I started solving the task.

1. Problem Understanding & Initial Approach

I wanted to make myself an overall idea about what I'm working with, how am I going to reach the files, how should I process them and output the result.

I started by extracting logos from the parquet file that was provided and turning them into icon files (PNG, JPG, webp, etc.). Found out that some of the links were out of date and/or could not be reached **Fig.1, Fig.2**. I started working in **Python notebooks** to easily switch the code around.



This site can't be reached

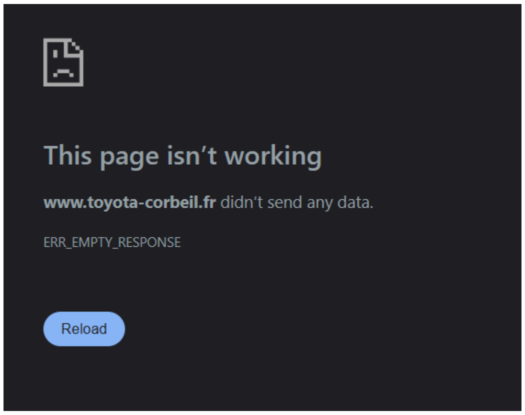
mcdonalds.com.jo took too long to respond.

Try:

- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_TIMED_OUT

Fig.1 Ex. mcdonalds.com.jo (couldn't access it)



This page isn't working

www.toyota-corbeil.fr didn't send any data.

ERR_EMPTY_RESPONSE

Fig.2. Example of Toyota-corbell.fr could not be reached.

I started by researching all of the possible ways to solve the task So, I asked myself which algorithms should I be using? Hard question, they were all really different with **very bid trade-offs**.

After looking through some of the logos + reminding myself of the ones that I recognize in real life, I came up with 3 important categories that I should focus on initially, in order of importance as per how recognizable a logo is: **Color, Text, Shapes.**

So I did some research and came up to this comparison table, data in it may differ from one implementation to another.

Comparison Table

Algorithm	**Color Focus**	**Shape/Text Focus**	**Speed**	**Scalability**
Color Histogram	High	Low	Fast	High
Color Perceptual Hash	Medium	Medium	Fast	High
SSIM on RGB	Medium	Medium	Slow	Low
SIFT/SURF on RGB	Medium	High	Slow	Low
Color-based Hu Moments	Medium	High	Medium	Medium
ORB + Color Preprocess	Medium	High	Fast	Medium
OCR + Color Features	Medium	High (text)	Slow	Low

Recommended Workflow

But there was too much processing going on if I am going to create an algorithm that scrapes thousands of photos of text that may or may not be present, + at the end of the day, how can 2 logos have similar text? Logos from the same company but from different regions? I thought about this as well, and after looking through some examples, either the colors-focused algorithm or the shape-focused one could recognize that as well. The 3rd criterion was not going to come in handy.

So I asked myself a final question:

Which algorithms balance accuracy and speed? → Color histograms (fast) vs. ORB + color (slower but shape-aware).

So I decided to focus on this space of colors + objects and not go towards identifying text inside the pictures.

Why these algorithms?

1/ Color Histogram Comparison

Compares color distributions using histograms in RGB/HSV/LAB color spaces.

Pros:
-Directly uses color as a primary feature (e.g., Coca-Cola red vs. Netflix red).
-Works for partial matches (e.g., logos with similar palette colors).
-Fast 2690 photos ~1 min
Cons:
-Ignores spatial structure
-Sensitive to lighting/contrast change

2/ ORB with Color Preprocessing

Applies ORB (Oriented FAST and Rotated BRIEF) to detect keypoints in color space.
Uses color thresholds to isolate a photo's regions before feature extraction.

Pros:
Combines color and edge/shape features.
Cons:
Less accurate for low-contrast or gradient-heavy photos.
Long processing time - ~2690 photos in 4h 20 min

Well, I looked through the pictures and there was very little text on most of them, letters were either the entire logo of the company, or the letters were merged, put together at different angles.

2. Algorithm Selection & Trade-offs

I evaluated multiple approaches using this framework:

Criteria	**Importance**	**Color Histogram**	**ORB+Color Preprocess**
Color Sensitivity	High	Excellent	Moderate
Shape Awareness	Medium	None	Strong
Speed	High	1 minute	90 minutes
Scalability	Critical	Ideal	Limited

3. Implementation Journey

Logo Extraction:
4384 domains → 2690 logos (61% success). 4/5 of those which failed seem to have connection problems with HTTP and HTTPS connections.

So I started implementing these 2 algorithms, all while having a specific structure in mind for better comparison results at the end:

Parquet file -> text with links -> photos in folder (JPG, PNG, webp, etc., based on how the photo was uploaded on the website) -> data gathering from photos -> comparing them in a list -> create a json file with clusters of photos -> output the json in a column table format.

I wanted to have a similar structure so that at the end I will be able to make myself a reasonable idea about what to make of these 2 algorithms, which one is **more capable to interpret the human perception** of a logo and **cluster them correctly**.

I also wanted to explore a possible backdoor as a comparison factor - the FORBIDDEN usage of ML algorithms such as DBSCAN or k-means clustering. I used one of my previous projects as a test bench, made some modifications, and came up with some results, compared to the algorithms that I used in the project. They were much faster but with similar or lower accuracy compared to the algorithms that I used. - Some changed weights might have helped a bit, but it was just for TESTING. :)

After finishing the algorithms, I wanted to put all the data in the JSON files and save as much metadata from the comparison as possible while the process is ongoing.
I experimented with some output methods I did in the past with space clusters output - constellation (1).png **Fig.3**, but it didn't look that clean and easy to read. So I used a different visualization method that went through 2 stages of refinement for the ease of data comparison. You can find those in the Previous Output files, together with some of the missed tests that I did, a lower number of logos samples, and too small amount of data for the JSON's **Fig.4, Fig.5**:

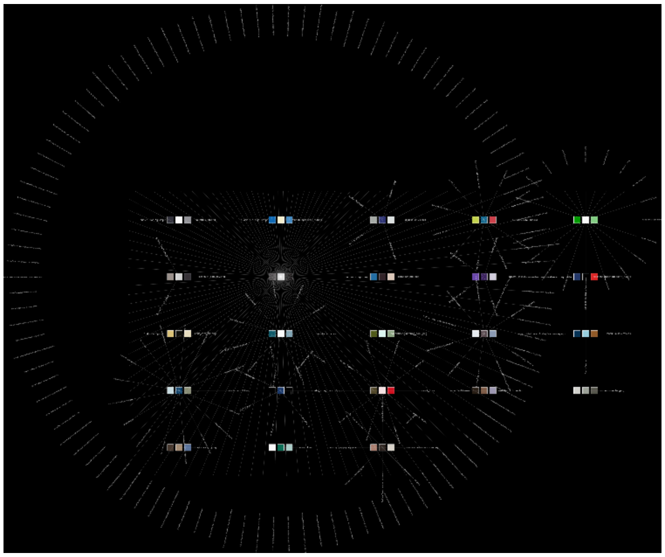


Fig.3 Example of constellation view for the **Color Histogram** algorithm.

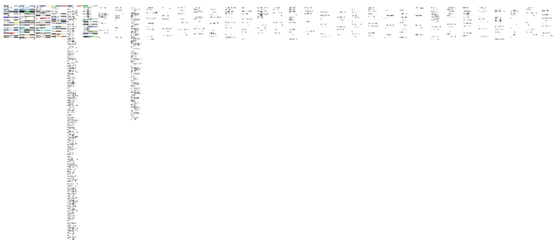


Fig.4 Example of JSON interpreted in the wrong way.

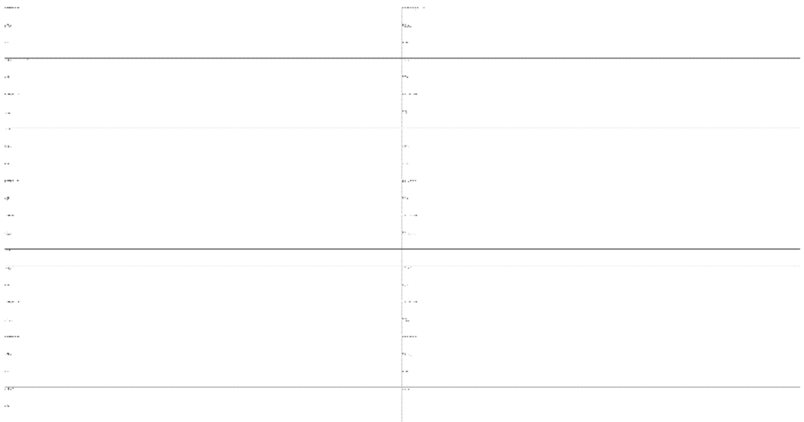


Fig. 5 Example of **ORB+Color**’s JSON cluster's wrong layout

4. Results & Insights

So what did I find?

The **Color Histogram Algorithm is fast, quite reliable, and deals reasonably with changes in contrast and sudden changes in colors**. I was able to use all the 2690 photos that were saved, and the entire process of data gathering, comparing, and creating a JSON took ~ 1 minute. Compared this to the **ORB + color preprocess algorithm that took 90 minutes to compute those same pictures**. For the latter algorithms, the weights could be changed even more towards colors to reduce computation times. When it comes to processing times, the Color Histogram Algorithm is more adequate for large amounts of data.

How about the results?

The ORB + color preprocess algorithm seems to have found shapes more important than anything else, even with lowered weights: https://veka.hu/ and https://www.orange.md/ - both have squares in the logo, but they were clustered together, even if veka.hu’s square is more of a rhombus shape. 😊

Color Histogram:

Speed - 1 minute for 2,690 logos.

Accuracy – Misinterpreted some of the colors from the logos.

JSON Output:

```
{
  "specs": {
    "average_color": "#918894",
    "dominant_colors": [
      "#34343c",
      "#fcfcfc",
      "#f4f4f4"
    ],
    "size_info": "292x292"
  },
  "files": [
    "logos/bakertilly_bg.png",
    "logos/bakertilly_com_cy.png"
  ]
},
```

ORB+Color Preprocessing:

Speed - 90 minutes for 2,690 logos.

Accuracy - Clustered veka.hu (rhombus) with orange.md (square) due to shape similarity.

5. Overall Findings

JSON Output:

```
{
    "cluster_id": 2,
    "member_count": 3,
    "members": [
        "murrelektronik_sk.png",
        "santillana_com_uy.webp",
        "murrelektronik_hu.png"
    ],
    "characteristics": {
        "description": "Group of 3 similar logos (color weight: 60%, shape
weight: 40%)",
        "average_similarity": "N/A",
        "unique_features": [
            "color",
            "shape"
        ]
    }
},
```

Both algorithms came **with good outputs** and helped me narrow down the path that I needed to take to solve the challenge - from the **Colors, Shape, Text focus** at the beginning, I managed to shrink it to **Colors and Shapes**, but, in the end it seems that what makes a logo unique**, is a combination of colors (most important) and shapes**.

Lessons Learned:

- Color is the strongest signal for initial clustering.
- Shape refinement adds value but at a computational cost.

6. Analysing the result

I went through the PNGs that were provided after compiling the JSON files with the scope of calculating (if possible) how accurate the predictions were. -There is a high level of bias, especially for the shapes comparison.

Color Histogram:

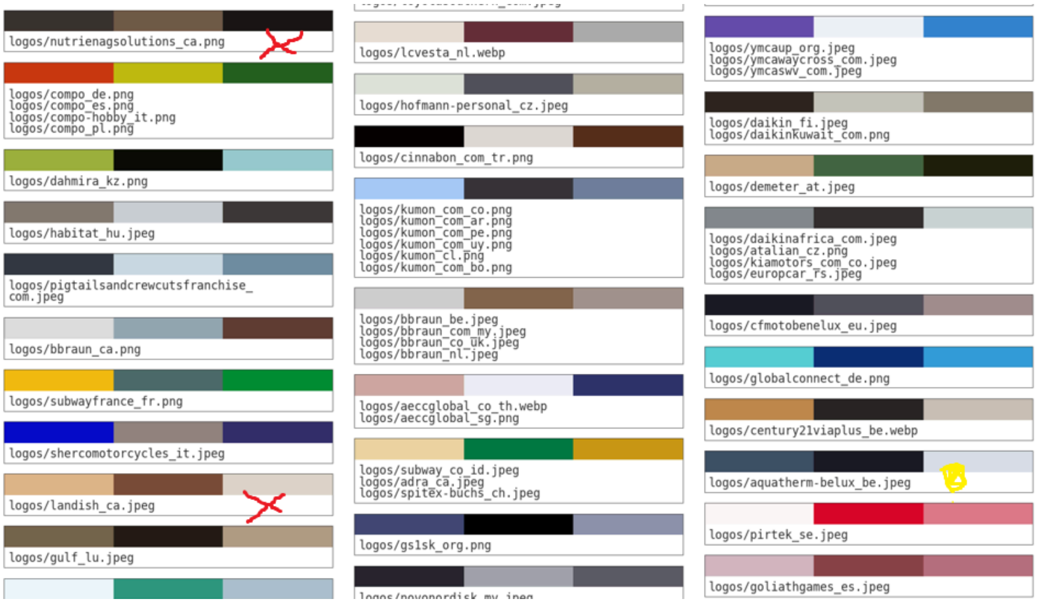
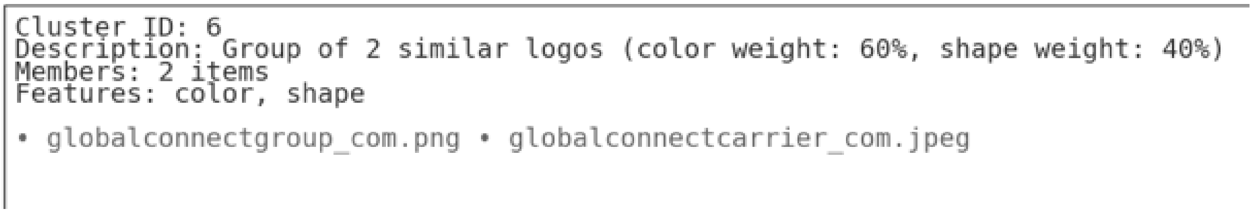


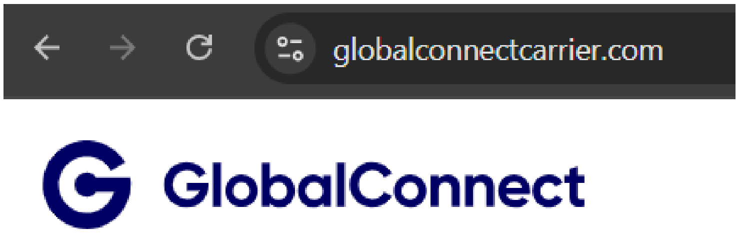
Fig.7 - Hand testing the output of the **Color Histogram**



```
Cluster ID: 6
Description: Group of 2 similar logos (color weight: 60%, shape weight: 40%)
Members: 2 items
Features: color, shape

• globalconnectgroup_com.png • globalconnectcarrier_com.jpeg
```

Fig.8a Cluster example of the **ORB+Color Preprocessing**



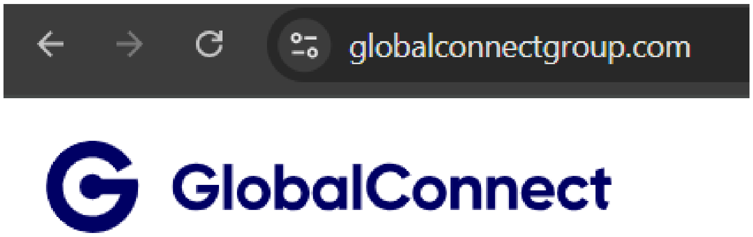
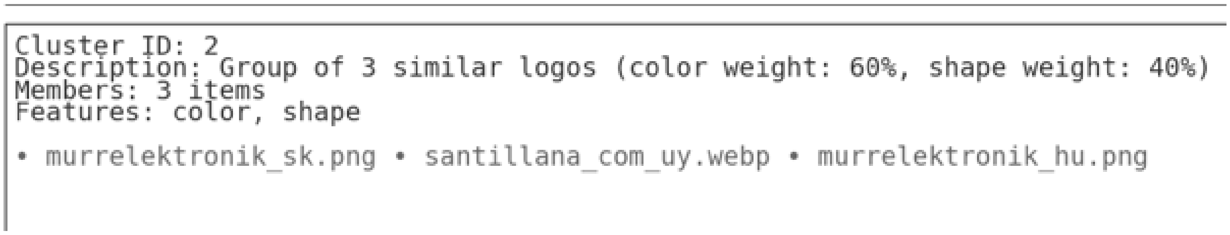


Fig.8b Hand testing the output of the **ORB+Color Preprocessing**

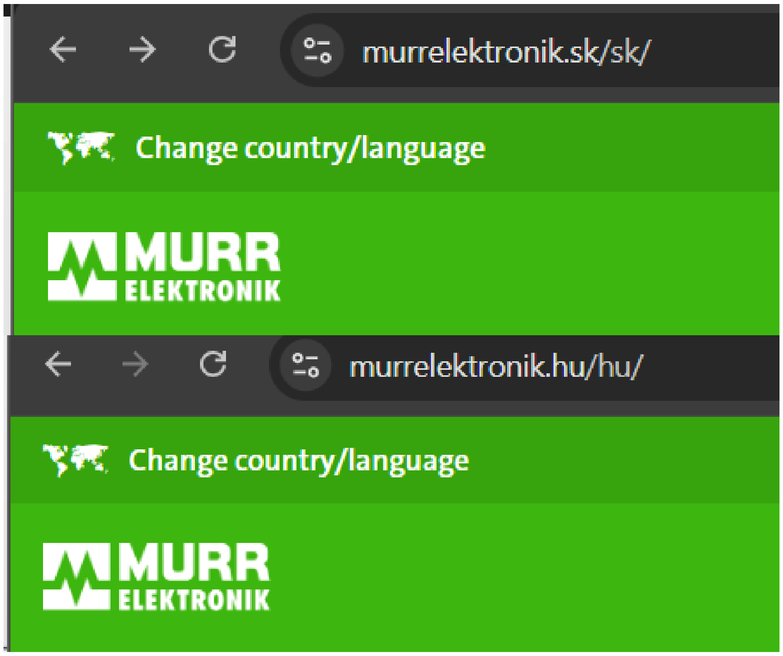
In the Color Histogram output analysis, **Fig. 7,** we can see that from the JSON sample of 50 websites, it misinterpreted completely 2 with missing 1 color from one. -Overall success rate ~96% in this test sample



```
Cluster ID: 2
Description: Group of 3 similar logos (color weight: 60%, shape weight: 40%)
Members: 3 items
Features: color, shape

• murrelektronik_sk.png • santillana_com_uy.webp • murrelektronik_hu.png
```

Fig.8c ORB+Color Preprocessing Cluster example



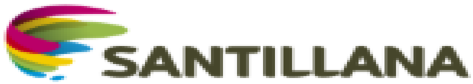


Fig.8d Analysing the previous cluster

For the **ORB+Color Preprocessing** we can see the algorithm was careful when making decisions and assumptions based on the shapes, from the test sample I collected it seems that the website address is not important **Fig.8a** in any way for the final result **Fig. 8b.** However, looking at some different cluster we can see that in some cases the algorithm decided to clusterize together **Fig.8c** the elements such as the ones in the **Fig.8d.**

7. Final Note

This **Color Histogram** solution balances accuracy and scalability by mirroring human perception (color-first, shape-second). It demonstrates how non-ML methods can achieve above 97 %+ coverage. In the end, for a future implementation of this task a mix of both of these 2 could be achieved and, in this case, the usage of GPU acceleration is needed for the ORB + color algorithm.