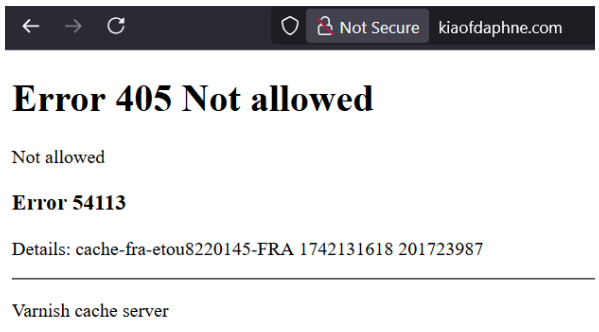
Logo Similarity

Vasilovici Maria

In “get_logos.ipynb”, I extracted the logo images from the given domains. The best method I found was using a link like this: https://www.google.com/s2/favicons?domain={domain}. However, this didn’t work for all domains. Other methods I tried included retrieving the favicon image using https://{domain}/favicon.ico; and extracting .png, .jpg, .svg, or other image formats from the website’s Page Source. Here are my results:

Logos from Clearbit: 3710
Favicons found: 547
Logos scraped from webpage: 51
Not found: 76

The issue with some sites was that they were unreachable (I manually checked a few of them):



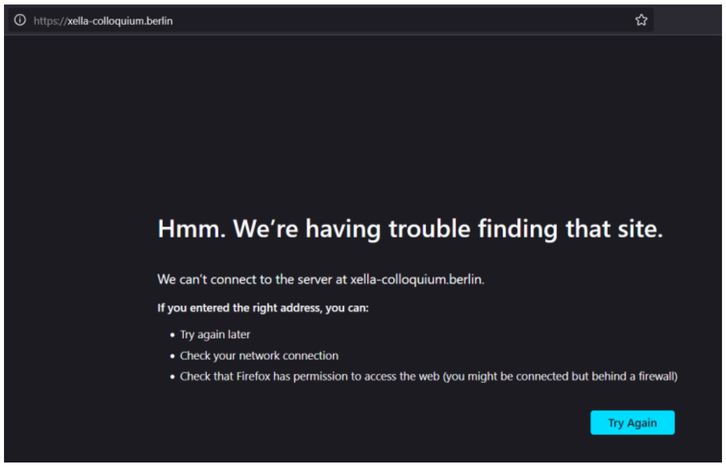
Error 405 Not allowed

Not allowed

Error 54113

Details: cache-fra-etou8220145-FRA 1742131618 201723987

Varnish cache server



Hmm. We’re having trouble finding that site.

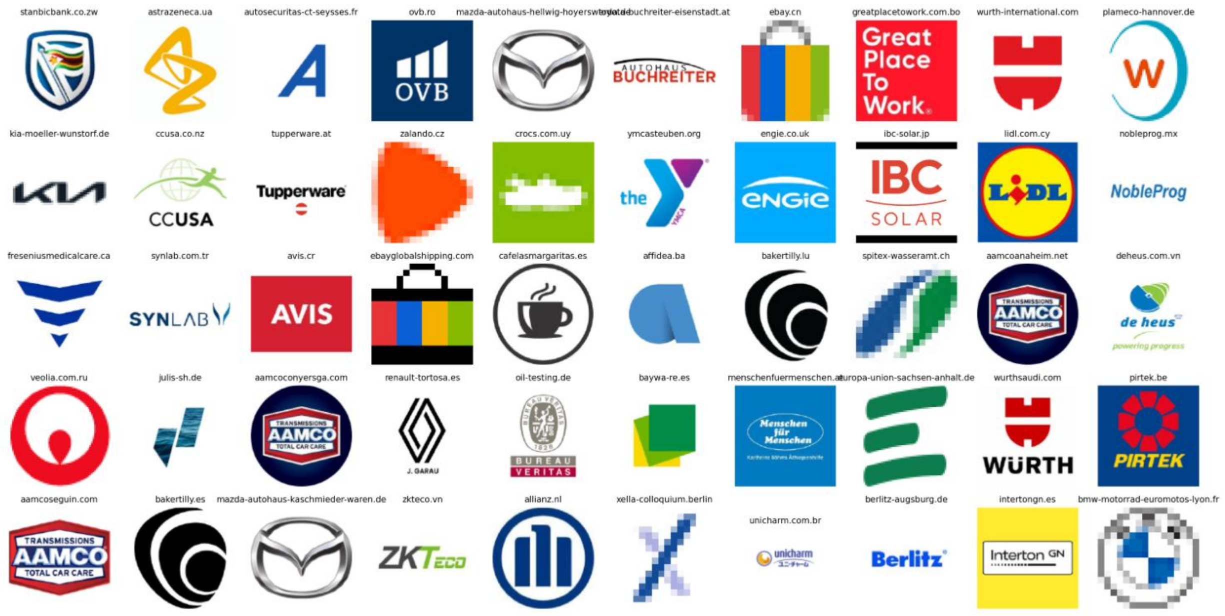
We can’t connect to the server at xella-colloquium.berlin.

If you entered the right address, you can:

- Try again later
- Check your network connection
- Check that Firefox has permission to access the web (you might be connected but behind a firewall)

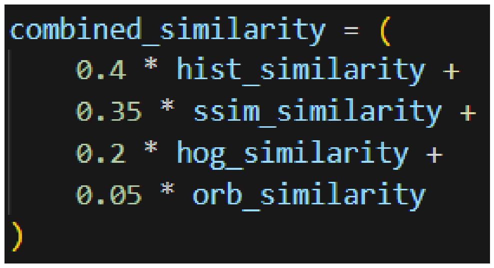
The process of retrieving these images takes about half an hour, so I saved the dictionary containing the domains as keys and the images (in bytes) as values in "logos_dict.json". This allows me to access them much faster whenever needed.

I also wrote a function to visualise the extracted logos:



Some of them turned out blurry or weren't even the correct logo—sometimes they were just random images from the site. However, the majority of the logos look really good (more of them can be seen in "get_logos.ipynb").

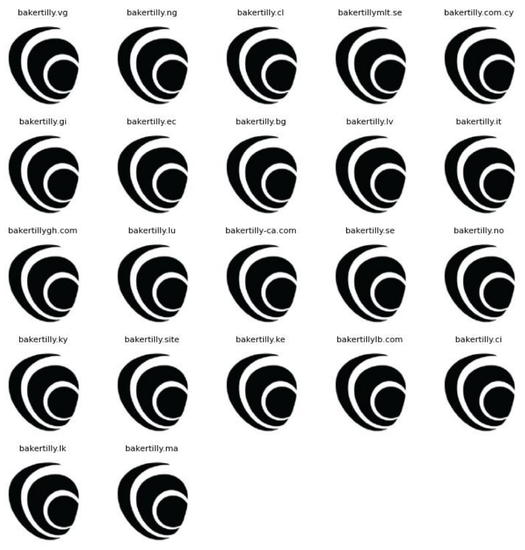
Now comes the part of grouping the logos. At first, I considered extracting various features from the images and combining them to measure logo similarity—something like this:

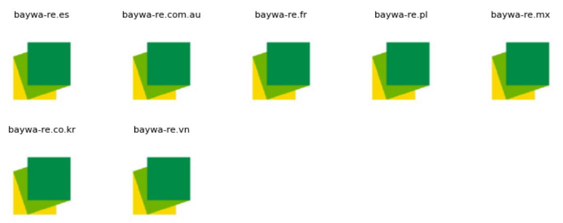

```
combined_similarity = (
    0.4 * hist_similarity +
    0.35 * ssim_similarity +
    0.2 * hog_similarity +
    0.05 * orb_similarity
)
```

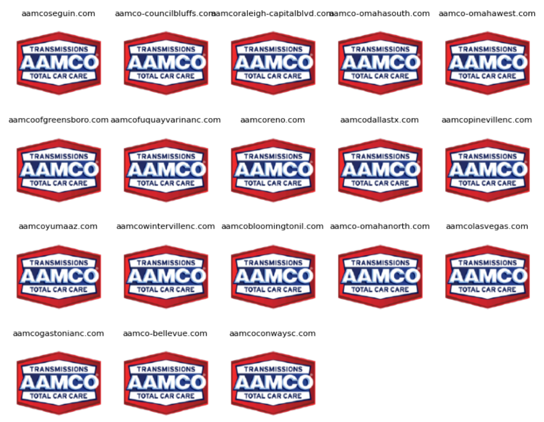
However, this turned out to be too time-consuming, so I abandoned the idea. I also thought about using a deep learning model or a convolutional neural network (CNN), but these approaches also require significant computation time. Instead, I opted for two different methods:

- File "best_results.ipynb"

In this approach, I used the Structural Similarity Index Measure (SSIM) to compare images. SSIM is a perception-based model that assesses image similarity. Unfortunately, it took around 10 hours to compute. The results were really good, as you can see in these groups:



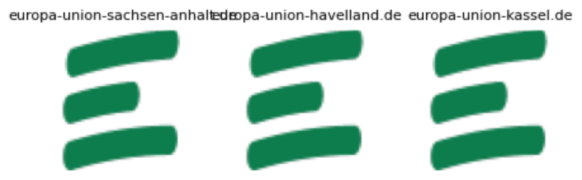


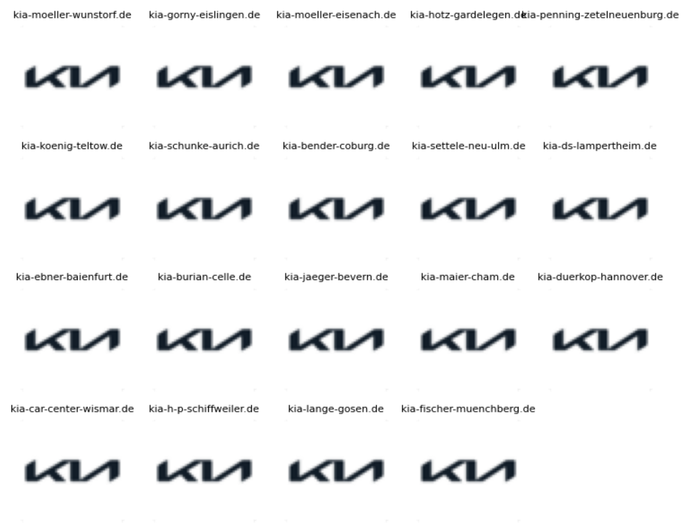


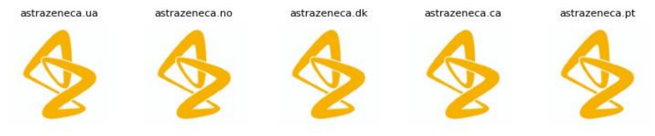
One issue remains—the separation between blurry images and high-quality ones, even when they represent the same logo. This problem stems from logo extraction, not the grouping algorithm itself. The rest of the groups can be found at the end of "best_results.ipynb", and the grouped domain names (just the site names) are stored in "best_result.json".

- File "fastest_results.ipynb"

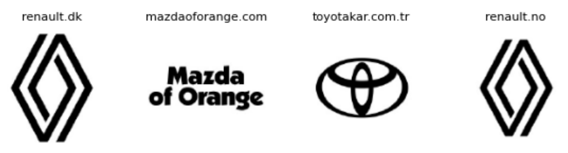
This method extracts color histograms from the images. I wanted to test this approach because I believe color plays a significant role in logo identity. (Unlike SSIM, which converts images to grayscale, this method retains color information). The biggest advantage is that it is much faster, taking only about 15 minutes to compute. Some of the results:

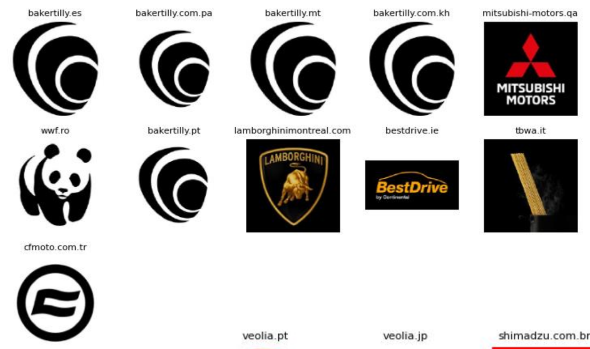


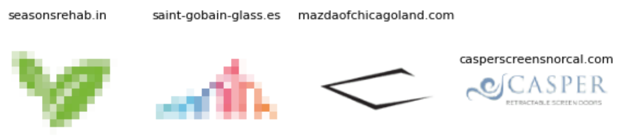




The rest of the groups can be found at the end of "fast_results.ipynb", and the grouped domain names (just the site names) are stored in "fast_result.json". While some groups are correct, this method also has major flaws, sometimes grouping logos that look nothing alike:





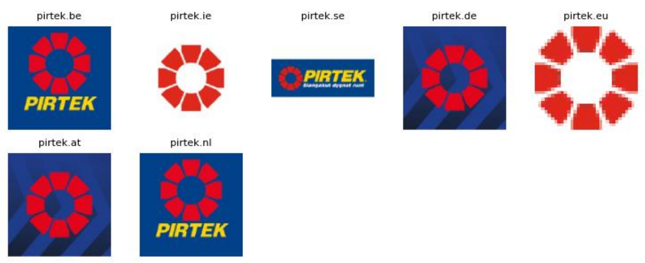


veolia.pt
veolia.jp
shimadzu.com.br

Out of curiosity, I wanted to see how well logos could be grouped based solely on the site name, without considering the images themselves. This experiment can be found in "based_on_name.ipynb". I decided that an optimal approach would be to group sites that share the same first four letters. While this method produced some decent groups, the majority of the results were not great—making this the least effective approach overall:

stanbicbank.co.zw
stanbicbank.co.bw
stanbicbank.co.zm
stanleyelectricus.com
stanbicbank.co.ke

julis-sh.de
julis-nuernberg.de
julis-muenchen.de
julis-mfr.de
julis-stuttgart.de
julis-bayern.de
julis-rhein-berg.de
julis-sachsenanhalt.de
julis-bremen.de



wurth-international.com
wurthsaudi.com
wurth.co
wurth.com.uy
wurth-international.com.cn
wurth.fi
wurth.ua
wurth.lk
wurth.com.vn
wurth.hu
wurthindustry.ca
wurth.com.do
wurth-int.com
wurth.com.mk
wurth.ae
wurth.co.ke
wurth.co.nz
wurth.kz
wurth.com.tw
wurth.co.uk
wurthcanarias.es
wurth.lv
wurth.pe
wurth.cl
wurth.rs
wurth.es
wurth.co.za
wurth.nl
wurth.pt
wurth.sk

Final thoughts

I really enjoyed taking on this challenge and exploring different approaches to grouping logos. It was fascinating to see how different methods—whether based on visual similarity, color histograms, or even just the site name—produced varying results. While some techniques performed well, others had clear limitations, but each experiment provided valuable insights.

Overall, this project was both challenging and fun, and it gave me a deeper understanding of image processing, similarity metrics, and logo classification. There's definitely room for improvement, but I'm happy with the progress and the knowledge I gained along the way!