VIETNAM UNIVERSITY
HO CHI MINH UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

FINAL LAB REPORT

FIND IMAGE INFORMATION OF LOGOS RELATED TO THE INPUT IMAGE

Subject: Visual Information Retrieval
Class: 20TGMT

Group 6 Members:

+ Dương Minh Tùng - 20127380
+ Đỗ Đạt Thành - 20127411

Lecturer & Instructor:

+ Teacher Võ Hoài Việt

Ho Chi Minh City - 18/08/2023

1. Member information: 2
2. Problem Definition: 2
Input and Output: 2
3. Ideas for solving the problem: 2
3.1. Query Image: 3
3.2. Feature Extraction: 3
3.2.1. Canny Edge Detection Preprocessing: 3
3.2.2. Feature Extraction using EfficientNetB7: 3
3.2.3. Enhanced Focus on Corner and Edge Features: 4
3.3. K-Means Clustering: 4
3.4. Similarity Comparison with Clusters: 4
3.5. Retrieval of Cluster Images: 4
3.6. Resultant Cluster: 4
3.7. YOLOv8: 4
4. Dataset: 5
4.1. Logo Dataset 5
4.2. LogoDet-3K 6
5. Experiments: 6
5.1 Model Installation: 6
5.2. Input images: 6
5.3. Prediction Function: 7
5.4 Feature Extraction: 7
5.5 Dataset Feature Extraction: 7
5.6 KMeans Clustering: 8
5.7 Retrieval Images 8
6. Self Evaluation: 11
7. References: 11

1. Member information:

Group 6 Members:

Name	ID	Work Done
Đỗ Đạt Thành	20127411	• YOLOv8 Train • Canny Edge Detection • Extract the feature through EfficientNetB7 • Label Clustering
Dương Minh Tùng	20127380	• Data Preprocessing for the YOLOv8 training • Documentation, Slides & Poster

2. Problem Definition:

Input and Output:

Input:

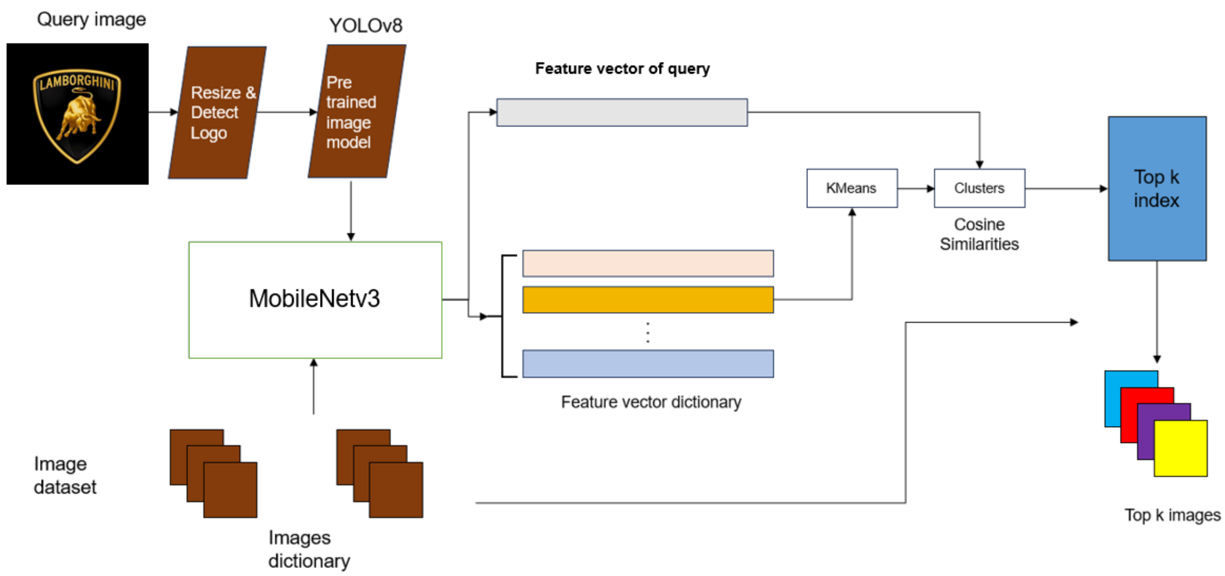
- The input to the system is an image that contains a logo. This image serves as the basis for the logo-related information retrieval process.
- The input image can vary in terms of its content, dimensions, quality, and logo placement.

Output:

- The desired output of the system is a top-k collection of images that are relevant to the logo depicted in the input images.
- These relevant images are retrieved from a database of images and are chosen based on their visual similarity and the similarity scores within the input image.

3. Ideas for solving the problem:

Model Architecture for Image-Based Logo-Related Information Retrieval



3.1. Query Image:

The query image is the input to the system. It contains a logo for which related images are to be retrieved from the database.

3.2. Feature Extraction:

To optimize the precision and specificity of our logo recognition and retrieval system, we have implemented a multi-step image preprocessing strategy. This methodology incorporates the use of the Canny edge detection algorithm prior to the feature extraction phase, now employing the MobileNetV3_small model.

3.2.1. Canny Edge Detection Preprocessing:

Before feature extraction, we apply the Canny edge detection algorithm to the input images. This technique significantly enhances the visibility of edges, corners, and contours present within the images. By accentuating these foundational elements, we provide the subsequent feature extraction process with a more refined and distinctive portrayal of the image.

3.2.2. Feature Extraction using MobileNetV3_small:

Building upon the Canny edge detection preprocessing step, the images undergo feature extraction using the MobileNetV3_small model. MobileNetV3_small is a cutting-edge convolutional neural network architecture specifically tailored for image classification tasks. This model excels in capturing intricate patterns and distinctive attributes embedded within the images.

3.2.3. Amplified Emphasis on Corner and Edge Features:

The fusion of Canny preprocessing with subsequent feature extraction using MobileNetV3_small establishes a symbiotic synergy. Consequently, the MobileNetV3_small model places enhanced emphasis on corner and edge features, which were amplified during the preprocessing phase. This heightened focus further augments the model's prowess in recognizing and differentiating logos based on these distinct structural attributes.

By embracing this two-pronged approach, we harness the collective strengths of advanced image preprocessing techniques and deep learning methodologies. This synergy effectively spotlights and captures pivotal features such as corners and edges, underpinning precise and accurate logo recognition and retrieval.

3.3. K-Means Clustering:

The K-means clustering algorithm is employed to group the extracted features into clusters based on their similarity. The clusters formed by K-means represent distinct visual patterns and attributes shared among logos in the database.

3.4. Similarity Comparison with Clusters:

Instead of running the query image features through K-means, the extracted features are directly compared with the feature clusters formed by K-means.
This comparison is facilitated by measuring the Cosine Similarity between the query image features and the cluster centroids.

3.5. Retrieval of Cluster Images:

The images associated with the clusters that exhibit the closest similarity to the query image features are retrieved from the database. These retrieved images share visual similarities with the logo in the query image, even if they belong to different categories or contexts.

3.6. Resultant Cluster:

The resultant cluster represents a set of images from the database that share visual characteristics with the logo in the query image. These images are deemed relevant and closely related to the logo present in the query image.

3.7. YOLOv8:
In our pursuit of robust and efficient logo detection, we have used YOLOv8, an advanced object detection model known for its exceptional speed and accuracy. YOLOv8, revolutionizes logo detection by providing real-time capabilities without compromising on precision. This model capitalizes on a single-pass architecture, enabling it to simultaneously identify logos and their precise locations within images. By leveraging YOLOv8's prowess, we are able to streamline logo detection, making it an integral component of our comprehensive image-based logo-related information retrieval system.

4. Dataset:
4.1. Logo Dataset
The training dataset for query-related tasks comprises an expansive collection of 1319 distinct brands, accompanied by an extensive compilation of approximately 88,000 images. This rich and diverse dataset forms the foundation for honing the capabilities of our logo recognition and retrieval system. With a considerable array of brands and images, our model benefits from a comprehensive representation of the logo landscape, empowering it to proficiently identify and retrieve logos across various contexts and scenarios. The amalgamation of such a substantial dataset and advanced methodologies positions our system to deliver superior performance in logo-related image retrieval.

4.2. LogoDet-3K
The LogoDet-3K dataset is organized into 9 overarching super categories, each encompassing several sub-categories, images, and annotated objects. The statistics of these super categories provide an insightful glimpse into the dataset's diverse content such as: Food, Clothes, Electronics, Necessities, Transportation, Sports… which are used for training YOLOv8.

5. Experiments:
5.1. Model Installation:
To establish our model, we initialize the model_path variable, which denotes the path to the trained model file (last.pt/best.pt). This model forms the backbone of our logo detection and retrieval system.

5.2. Input images:

In the beginning, we will input in an image of Lamborghini Logo, then we will detect the logo on the the picture and we have a result like this:

```
#Detect logo in the image
row, column = pred("/content/lamborghini.jpg")
#Crop logo detected and save as file
crop_save("/content/lamborghini.jpg",row,column)
```

```
0: 384x640 1 logo, 94.1ms
Speed: 24.1ms preprocess, 94.1ms inference, 46.5ms postprocess per image at shape (1, 3, 384, 640)
```

The input logo will be cropped and saved as a new file called *cropped.jpg,* then we will use that image file for retrieval of the top-k relevant images.

5.3. Prediction Function:

Our prediction function takes img_path as input, indicating the path to the image to be searched. It then generates an output image with selected regions highlighted if the prediction confidence exceeds a defined threshold as 0.5. These highlighted regions denote potential logo detections.

5.4. Feature Extraction:

The images, having undergone enhancement, proceed to feature extraction. The selected model, 'MobileNetv3_small,' is invoked, operating either on CPU or GPU, based on the available hardware.

5.5. Dataset Feature Extraction:

Our system reads and processes the dataset, extracting distinctive features. The *dataset_path* variable guides the system to the dataset directory (e.g., './data_extract').

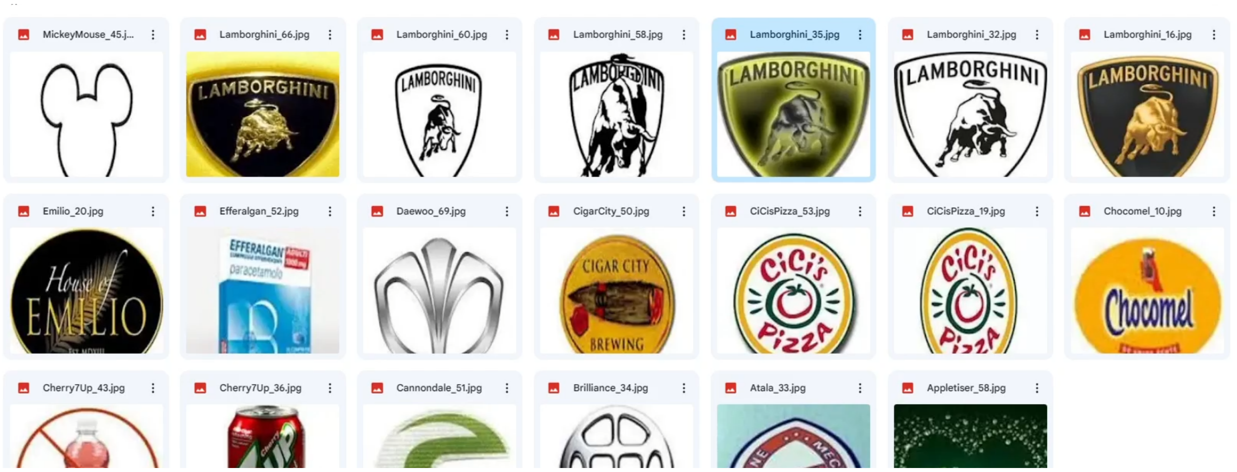
5.6. KMeans Clustering:

We employ the KMeans algorithm to cluster images with similar feature vectors in the dataset, assigning labels for reference.

5.7. Retrieval Images

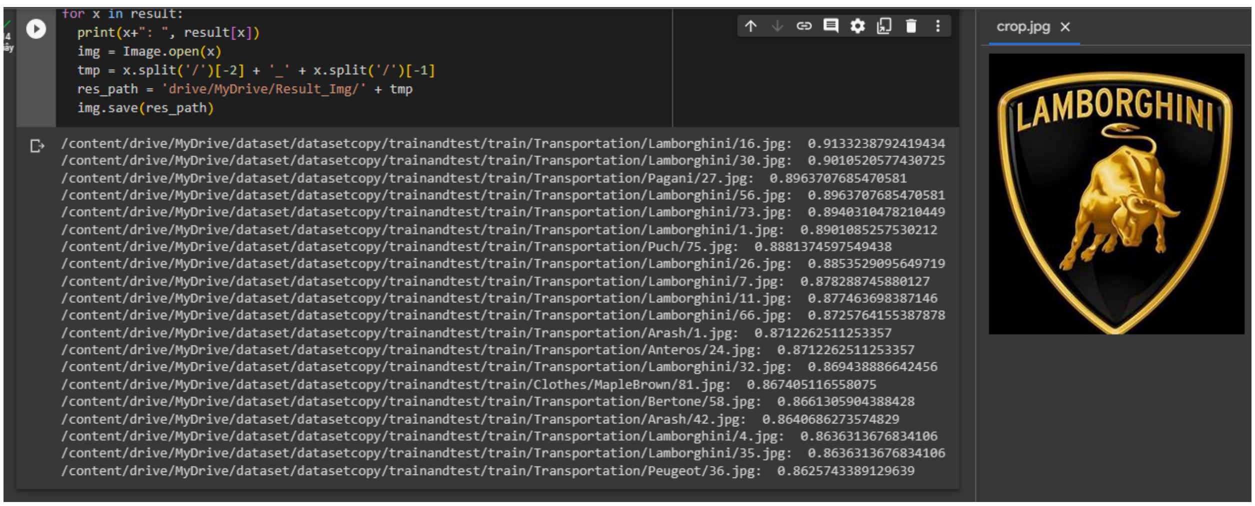
Upon the completion of the detection phase, the system takes as input the path to the cropped image and the user-specified number of top similar images denoted as top_k. Subsequently, the system initiates the process of predicting the label associated with the input image. It then proceeds to evaluate the similarity between the input image and the dataset images that share the same label, leveraging the robust CosineSimilarity algorithm available within the torch library. The obtained similarity scores are meticulously organized in descending order, effectively ranking the images based on their resemblance to the input. From this sorted collection, the top k similar images are singled out, ensuring that the most pertinent matches are identified.

To enhance user experience and organization, the system dynamically assesses the presence of the result directory. If this directory is non-existent, it is created. Alternatively, if it is already present, the directory is cleared, ensuring that previous results do not confound the current output. To offer further insight into the comparison process, the system unveils the similarity scores associated with each dataset image. Additionally, the system diligently archives the retrieved images in the designated result directory, solidifying the culmination of our logo-related image retrieval process. This holistic approach exemplifies the systematic and meticulous manner in which our system effectively combines prediction, similarity assessment, result organization, and result presentation to achieve a refined and intuitive logo-related information retrieval system. And the result as we have:

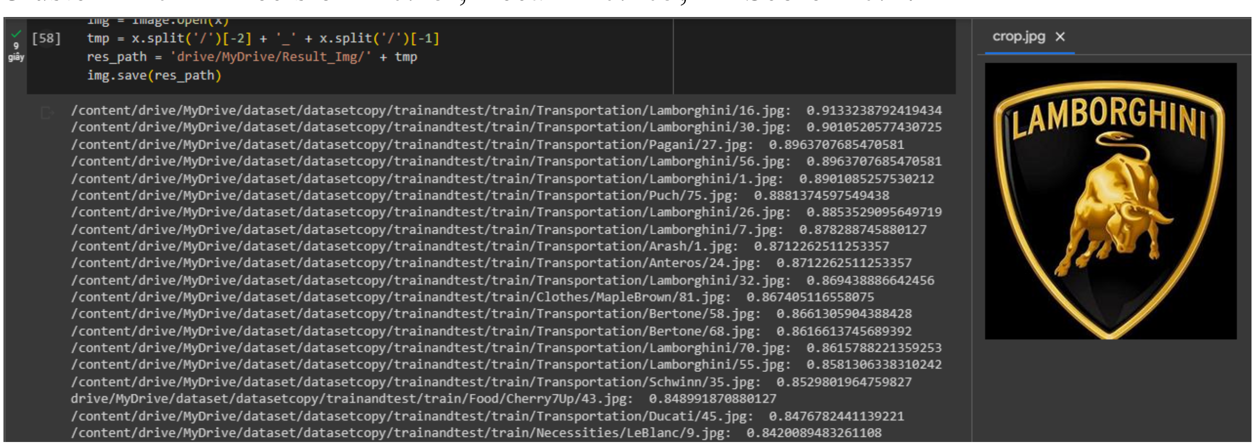


We also try with 4 clusters which are 5, 10, 15, 20 which given the result as below

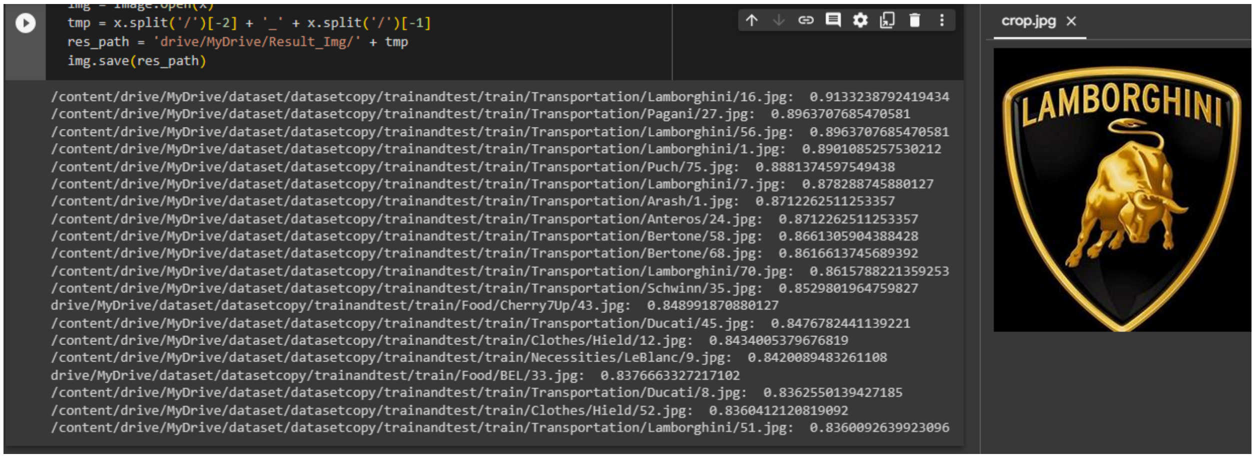
Cluster = 5 → Precision = 0.6 , Recall = 0.14, F1-Score = 0.23



Cluster = 10 → Precision = 0.45 , Recall = 0.105, F1-Score = 0.17



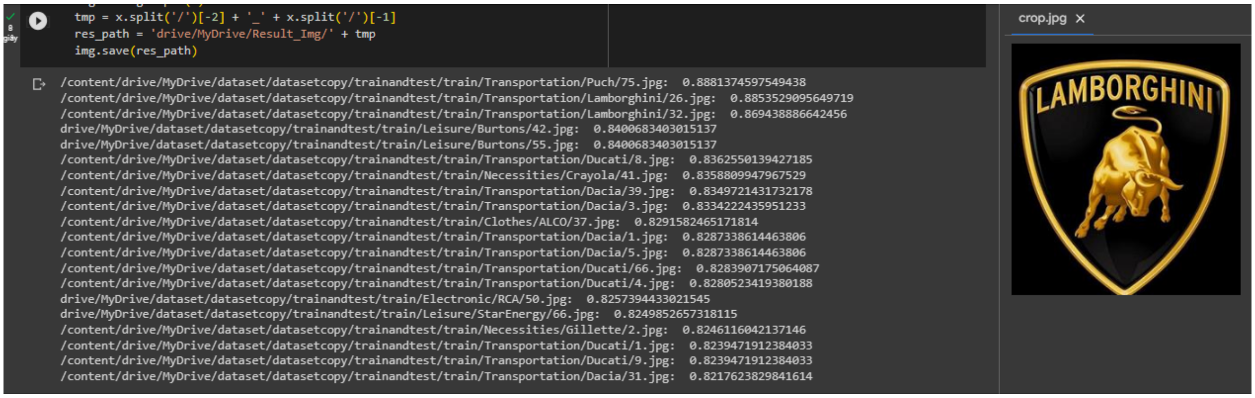
Cluster = 15 → Precision = 0.3 , Recall = 0.07, F1-Score = 0.11




```
    tmp = x.split('/')[-2] + '_' + x.split('/')[-1]
    res_path = 'drive/MyDrive/Result_Img/' + tmp
    img.save(res_path)
```

```
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Lamborghini/16.jpg:  0.9133238792419434
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Pagani/27.jpg:  0.8963707685470581
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Lamborghini/56.jpg:  0.8963707685470581
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Lamborghini/1.jpg:  0.8901085257530212
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Puch/75.jpg:  0.8881374597549438
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Lamborghini/7.jpg:  0.878288745880127
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Arash/1.jpg:  0.8712262511253357
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Anteros/24.jpg:  0.8712262511253357
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Bertone/58.jpg:  0.8661305904388428
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Bertone/68.jpg:  0.8616613745689392
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Lamborghini/70.jpg:  0.8615788221359253
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Schwinn/35.jpg:  0.8529801964759827
drive/MyDrive/dataset/datasetcopy/trainandtest/train/Food/Cherry7Up/43.jpg:  0.848991870880127
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Ducati/45.jpg:  0.8476782441139221
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Clothes/Hield/12.jpg:  0.8434005379676819
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Necessities/LeBlanc/9.jpg:  0.8420089483261108
drive/MyDrive/dataset/datasetcopy/trainandtest/train/Food/BEL/33.jpg:  0.8376663327217102
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Ducati/8.jpg:  0.8362550139427185
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Clothes/Hield/52.jpg:  0.8360412120819092
/content/drive/MyDrive/dataset/datasetcopy/trainandtest/train/Transportation/Lamborghini/51.jpg:  0.8360092639923096
```

Cluster = 20 → Precision = 0.10 , Recall = 0.02, F1-Score = 0.03



Cluster	Precision	Recall	F1-Score
n = 5	0.6	0.14	0.23
n = 10	0.45	0.105	0.17
n = 15	0.3	0.07	0.11
n = 20	0.10	0.02	0.03

As the number of clusters increases, the accuracy of the system tends to decrease, subsequently diminishing the quality of the output. This phenomenon can be attributed to the inherent trade-off between cluster granularity and accurate representation. When clusters become too numerous, there is a higher likelihood of overfitting, wherein the model begins to capture noise and individual variations rather than meaningful patterns. This results in a less coherent and more fragmented grouping of images, leading to reduced accuracy in similarity assessments. Therefore, striking a balance between cluster quantity and quality is pivotal in achieving a harmonious interplay between accuracy and the meaningful organization of retrieved images.

6. Self Evaluation:

6.1. Incomplete Evaluation System:

One notable limitation of our system is the absence of a comprehensive evaluation mechanism. While our approach demonstrates promising results, the lack of a robust evaluation framework hinders our ability to precisely gauge its performance across diverse scenarios. To address this, we are considering the implementation of a validation strategy, either by augmenting the existing dataset with a dedicated validation subset or by sourcing an external dataset suitable for validation purposes.

6.2. Dependence on Algorithmic Approach:
A challenge we encounter is the need to avoid relying solely on personal assumptions or intuition when selecting datasets. To circumvent this limitation, we are emphasizing the integration of algorithmic methodologies and structured schema. This entails employing sophisticated algorithms alongside well-defined schemas for comparison, ensuring that our decisions are founded on accurate and objective analyses rather than subjective judgment.

6.3. Precision Limitation with Ambiguous Inputs:
Our system's accuracy faces limitations when processing inputs lacking distinct corners or edges, similar to shapes such as 'Mickey Mouse' or circular designs. This shortfall stems from the reliance on corner and edge features for accurate recognition. To enhance this aspect, we are exploring the integration of feature extraction tailored to specific shapes. By identifying and leveraging shape-specific features, we aim to bolster the system's accuracy even in cases of less defined geometries.

In acknowledgment of these limitations, we remain dedicated to refining and advancing our approach. The pursuit of a comprehensive evaluation system, the judicious use of algorithmic assessments, and the augmentation of feature extraction techniques underscore our commitment to addressing and overcoming these challenges, thereby elevating the overall performance and reliability of our logo recognition and retrieval system.

7. References:

- Open Set Logo Detection and Retrieval
- Logo Images Dataset
- LogoDet-3K
- PyTorch documentation — PyTorch 2.0 documentation
- MobileNetv3
- OpenCV: Canny Edge Detection
- Home - Ultralytics YOLOv8 Docs