POLYTECHNIQUE
MONTRÉAL

INF6804 Computer Vision

Project
Practical Assignment 1

Mohamed Aziz Younes - 2417117
Mathieu La Brie - 2403325

Polytechnique Montreal
2025-02-06

Contents

1 Presentation of the two compared approaches

1.1 Introduction

Describing regions of interest in images is important to many computer vision tasks, including object detection and content-based image retrieval (CBIR). Two distinct approaches for achieving this are Histograms of Oriented Gradients (HOG) and Contrastive Language-Image Pretraining (CLIP). While HOG is a traditional feature descriptor focusing on structural patterns, CLIP uses deep learning to understand content. This section compares these two methods in terms of their principles.

1.2 Histograms of Oriented Gradients (HOG)

The Histogram of Oriented Gradients (HOG) is a widely used feature descriptor in the fields of computer vision and image processing. It examines the distribution of edge orientations within an object to characterize its shape and visual characteristics. The HOG technique entails calculating the gradient magnitude and orientation for every pixel in an image, followed by segmenting the image into smaller cells.

1.2.1 description and principals

Preprocessing

First of all, HOG needs the image to be preprocessed with a width-to-height ratio of 1:2 and resized to 64x128, and that is because this specific size allows the image to be evenly divided into 8x8 and 16x16 patches for feature extraction. Using these dimensions simplifies the calculations and ensures consistent feature representation across the image.

Figure 1: Image resizing

The next step involves calculating the gradient for each pixel in the image. Gradients represent the small variations in the x and y directions. In this instance, I will extract a small patch from the image and compute the gradients within that area.

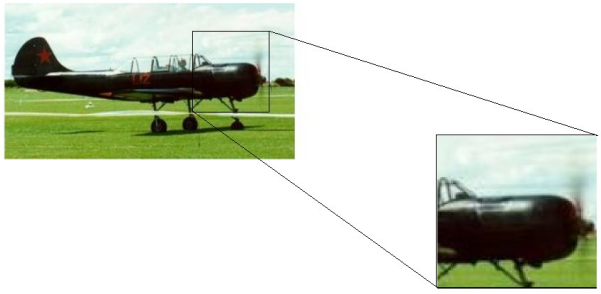

Figure 2: Patch extraction

Gradient Calculation

The HOG process starts by transforming the input image into grayscale, which streamlines the analysis by concentrating on intensity values. Gradients, indicating variations in intensity, are calculated for each pixel. These gradients emphasize the image's edges and textures, offering crucial insights into its structural patterns.

Orientation Binning

Next, the computed gradient orientations are grouped into discrete bins. For instance, a typical setup might divide the range from 0Â° to 180Â° into nine bins. The image is further divided into small regions called cells, often 8x8 pixels in size. For each cell, a histogram of gradient orientations is created, quantifying the distribution of edges within that region.

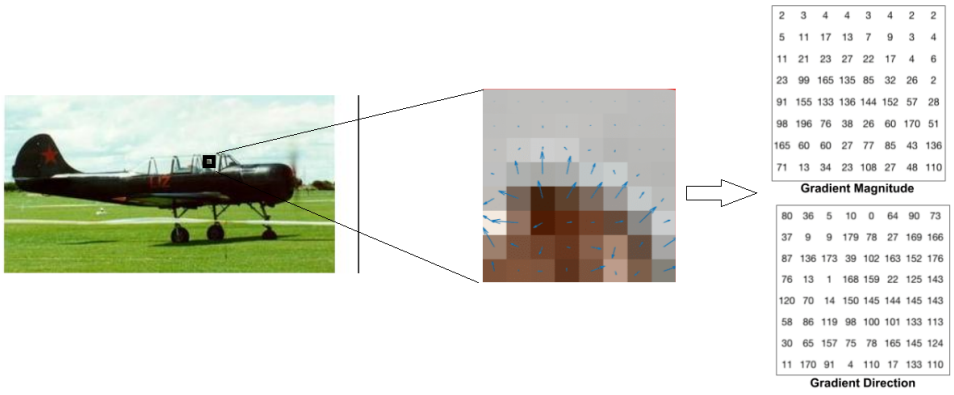

Figure 3: Gradient and orientation calculation

The next step is to create a histogram of gradients in these 8 by 8 cells. The histogram contains 9 bins corresponding to angles 0, 20, 40 to 160. The following figure illustrates the process. We are looking at magnitude and direction of the gradient of the same 8 by 8 patch as in the previous figure.

A bin is selected based on the direction, and the vote (the value that goes into the bin) is selected based on the magnitude. Letâs first focus on the pixel encircled in blue. It has an angle (direction) of 80 degrees and magnitude of 2. So it adds 2 to the 5th bin. The gradient at the pixel encircled using red has an angle of 10 degrees and magnitude of 4. Since 10 degrees is half way between 0 and 20, the vote by the pixel splits evenly into the two bins.

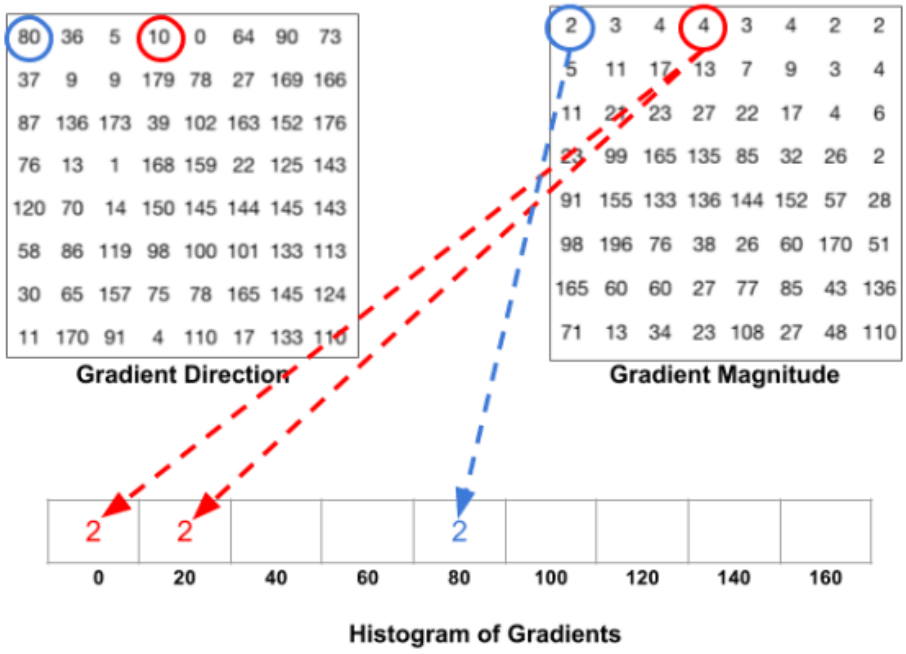

Figure 4: histogram of gradients creation

Block Normalization

To enhance robustness against variations in lighting and contrast, cells are grouped into overlapping blocks, such as 2x2 cells. The histograms within these blocks are normalized, ensuring consistency and stability across the image. This step is crucial for improving the descriptor's reliability in real-world scenarios.

Feature Descriptor

Finally, the normalized histograms from all blocks are concatenated to form a single feature vector. This vector serves as the HOG descriptor, a compact representation of the image's structural features.

Figure 5: output of HOG discriptor

1.3 Contrastive Language-Image Pretraining (CLIP)

CLIP is a deep learning model that aligns images and textual descriptions into a shared space, enabling tasks like image description, retrieval, and classification.
This model can understand both text descriptions and images by utilizing a training method that focuses on contrasting pairs of text and images.

1.3.1 description and principals

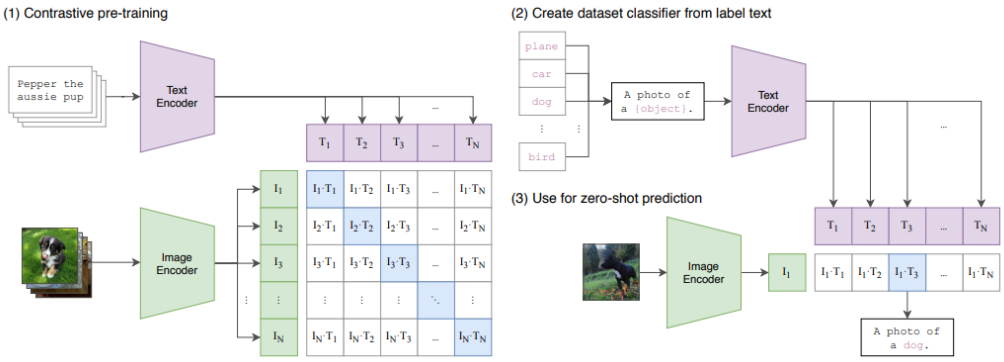

Figure 6: CLIP architecture [1]

Training

CLIP begins with a pretraining phase on a massive dataset of image-text pairs. Using contrastive learning, the model is trained to associate corresponding image-text pairs while distinguishing unrelated ones. For example, an image of a dog paired with the caption A brown dog running in a park is learned to be similar, whereas it is contrasted against unrelated captions like A red car on a road. This process ensures that CLIP learns meaningful associations between images and their textual descriptions, building a robust foundation for downstream tasks.

Feature Extraction

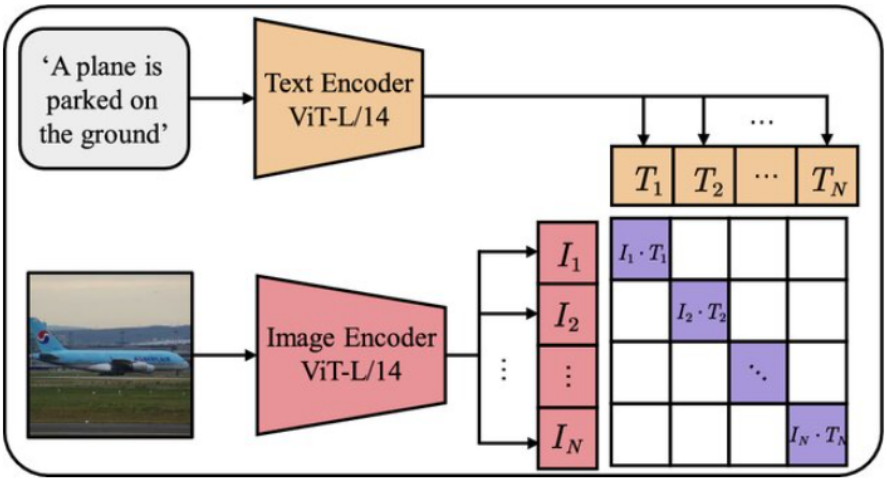

Figure 7: Text and image encoders [2]

Vision Encoder: Images are passed through a vision encoder, such as a Vision Transformer (ViT) or ResNet, which extracts visual features and represents them in a high-dimensional vector format.

Text Encoder: Similarly, textual input is processed by a Transformer-based text encoder, transforming the text into a vector representation.

Embedding

Once the image and text features are extracted, they are mapped into a shared latent space. In this space, similarity between images and text is measured using distance calculation methods like Ecludien distance or cosine simularity. The goal is to place related images and text closer together while keeping unrelated pairs farther apart. For example, the vector representation of an image of a cat will be closer to the text âA cute cat sitting on a sofaâ than to unrelated descriptions like âA bird flying in the sky.â

2 Performance hypotheses in specific use cases

2.1 Use case 1: Sensitivity to noise in the image

HOG is more sensitive to noise in the image due to its reliance on detecting local gradients, which correspond to edges or intensity changes in an image. Since HOG computes gradients for small patches, any random variations in pixel intensity caused by noise can introduce spurious gradients, which may be misinterpreted as meaningful edges. For example, in a clean image of an airplane,

HOG detects strong, structured gradients at the edges of the wings and fuselage. However, adding noise, such as Gaussian, can create false gradients in areas where there are no actual edges, leading HOG to mistakenly identify noise-induced patterns as part of the object.
In contrast, CLIP is more robust to noise because it learns high-level, semantic features rather than relying on pixel-level gradients. CLIP captures global features, such as the overall shape and context of an object, allowing it to generalize across noisy images. Since CLIP is trained on large, diverse datasets and combines image and text data, it can maintain object recognition even in noisy or cluttered environments, making it less sensitive to random pixel-level variations.

2.2 Use case 2: Sensitivity to Object Structure.

HOG and CLIP are both powerful methods for image retrieval, but they perform differently based on the nature of the object in the image. HOG is designed to capture local gradient information and is particularly sensitive to sharp edges and contours, making it ideal for structured objects like airplanes. The distinct features of an airplane, such as its wings, fuselage, and tail, create clear gradient changes that HOG can effectively capture. However, for smooth, featureless objects like balls, HOG struggles because their lack of sharp edges results in subtle or sparse gradient variations, leading to poor performance.
On the other hand, CLIP, which uses semantic associations between images and textual descriptions, is more robust to this challenge. CLIP recognizes objects based on their high-level, global features like shape, texture, and context rather than relying on edge information. This enables CLIP to perform well with both airplanes and balls, as it can understand the general shape of a ball or the concept of an airplane through semantic context, rather than depending on distinct gradients.

3 Performance hypotheses concerning the bounding boxes

The availability of bounding boxes for objects of interest in images is expected to enhance the performance of both HOG and CLIP in different ways. For HOG, bounding boxes can significantly improve accuracy by narrowing the focus to the object itself, eliminating background noise that might otherwise interfere with gradient-based feature extraction. For example, when detecting pedestrians, bounding boxes help isolate the human silhouette, ensuring that the gradients are calculated only for the relevant structure. Similarly, in traffic sign detection, bounding boxes prevent the inclusion of extraneous details, allowing HOG to focus on the distinct shapes and edges of the signs. This results in improved accuracy and reduced computation time, as smaller regions are analyzed.

In the case of CLIP, bounding boxes refine the semantic understanding of an image by reducing distractions from unrelated elements in the scene. For tasks such as content-based image retrieval or complex scene analysis, this focused approach enables the model to align image embeddings more

closely with textual descriptions. For example, if the query is âa red car,â bounding boxes ensure that the model processes only the car, avoiding unrelated objects like trees or buildings in the background. While this refinement enhances precision, the computational cost for CLIP remains largely unaffected, as it processes the image patch regardless of its size.

In summary, bounding boxes benefit both HOG and CLIP by concentrating their analysis on the most relevant regions of an image. HOG sees substantial gains in accuracy and efficiency due to its reliance on localized gradients, while CLIP achieves improved precision in embedding alignment, particularly in complex or cluttered scenes.

4 Description of experiments, data and evaluation criteria

4.1 Use case 1: Sensitivity to noise in the image

To test the sensitivity of HOG and CLIP to noise, we conducted an experiment where Gaussian noise with a mean of 0 and a standard deviation of 25 was added to both the query image and the images in the database. The query images were chosen from the dataset, and their HOG and CLIP features were extracted.
HOG, which focuses on local gradient patterns and edge information, was expected to be more sensitive to the noise, as random pixel variations interfere with the gradient detection process. CLIP, however, captures high-level semantic features, making it more robust to noise. For both clean and noisy query images, we computed the Euclidean distance between the query image descriptors and the database descriptors.
The top-5 retrieval accuracy was used as the evaluation metric. The results showed that HOG's performance significantly degraded with noisy images, as noise disrupted the gradient-based feature extraction. In contrast, CLIP maintained stable retrieval performance, demonstrating its resilience to noise due to its ability to capture global, semantic features.

Figure 8: Gaussian noise

CLIP proved to be more resilient to noise, delivering better retrieval results compared to HOG, especially in the presence of image degradation. This highlights CLIP's advantage in real-world applications where images are often subject to noise and variations in appearance.

4.2 Use case 2: Sensitivity to Object Structure.

For sensitivity to object structure, the experiments were designed to test how well HOG and CLIP descriptors could retrieve images of structured and unstructured objects, such as airplanes and balls, respectively, from the database. The hypothesis was that HOG would perform well with structured objects, such as airplanes with sharp edges and well-defined contours, but poorly with unstructured objects like balls, which lack strong gradients and textures. CLIP was hypothesized to handle both types of objects better due to its semantic understanding and ability to abstract features.

The primary difficulty with the query images in this context was that unstructured objects like balls have smooth, uniform shapes with few distinctive features, making it challenging for gradient-based descriptors like HOG to identify meaningful patterns. In contrast, structured objects such as

airplanes, with well-defined edges and textures, aligned more closely with HOG's strengths.

To measure similarity between descriptors, we again employed the Euclidean distance. This allowed us to compare the feature vectors of the query images with those of the database images and rank the database images based on their closeness to the query.

For evaluation, we used the top-5 retrieval accuracy as the metric, assessing how many of the retrieved images matched the query object type. This metric was applied separately to structured and unstructured queries, highlighting the differences in performance between HOG and CLIP for these object types.

4.3 Performance hypotheses concerning the bounding boxes

To test the performance hypotheses concerning bounding boxes, experiments were designed to analyze the ability of HOG and CLIP descriptors to retrieve images when the query objects were represented by bounding boxes. The hypothesis was that HOG, being sensitive to precise object shapes and gradients, would be more impacted by variations in the bounding box size and alignment, while CLIP, with its semantic and context-aware capabilities, would perform more robustly despite such variations.

The object in both the dataset and the query images are supposed to be manually identified and put in their rspective bounding boxes to achieve the experiment. To do this we reffered to Roboflow as a quick solution to give our images their bounding boxes and retrieve the bounding boxes cooredinates to later use them in our CLIP and HOG models.

The primary difficulty with using bounding boxes was maintaining consistent object representation across the dataset. Variations in the size, position, or alignment of bounding boxes could lead to cropped or incomplete representations of objects, potentially omitting critical features. This issue posed a challenge for HOG, which relies on capturing local gradients, as missing parts of an object could significantly alter the extracted features. CLIP, on the other hand, was expected to be less sensitive to such variations due to its ability to derive global and semantic representations.

The Euclidean distance was chosen as the similarity measure for comparing the descriptors extracted from the query and database images. This metric allowed us to compute the closeness of feature vectors, enabling the ranking of database images based on their similarity to the query.

The top-5 retrieval accuracy was used as the evaluation metric to determine how well each method performed in retrieving images that matched the query objects within bounding boxes. This metric assessed whether the correct objects appeared in the top five retrieved results, providing a clear comparison of the robustness of HOG and CLIP in handling bounding box-based queries.

By comparing the results, it was observed that CLIP maintained higher accuracy than HOG when variations in bounding box representation were introduced, validating the hypothesis that CLIP is more resilient to bounding box inconsistencies.

4.4 Compare the performance of different model sizes of CLIP

Summary of Results:

	Model	Patch Size	Average Similarity	Average Computation Time (s)
0	ViT-B/32	224	0.848176	0.170468
1	ViT-B/32	128	0.853844	0.159419
2	ViT-B/32	64	0.870377	0.245367
3	ViT-B/16	224	0.837642	0.432935
4	ViT-B/16	128	0.846126	0.381646
5	ViT-B/16	64	0.905478	0.430541
6	ViT-L/14	224	0.808431	1.907808
7	ViT-L/14	128	0.840199	1.936355
8	ViT-L/14	64	0.835348	1.874007

Figure 9: CLIP sizes comparison

decreasing the patch size leads to higher average similarity scores, suggesting that smaller patches may help capture finer image details, thereby improving feature alignment across images. This trend is most evident in the ViT-B/16 model, where the similarity score increases from 0.8376 at 224 resolution to 0.9055 at 64. However, reducing patch size does not always result in a linear increase in similarity, as seen in ViT-L/14, where the 64 patch size yields a slightly lower similarity than 128. In terms of computation time, larger models like ViT-L/14 take significantly longer to process images, confirming that model complexity increases processing cost.

5 Description of the implementations

The implementation of the Histograms of Oriented Gradients (HOG) method was carried out using the hog function from the skimage.feature library. The process involved converting input images to grayscale, calculating gradient magnitudes and orientations, and generating histograms of these orientations for small regions (cells) within the image. These histograms were normalized over overlapping blocks of cells and concatenated into a feature vector for each image. The primary parameters included orientations (set to 8), pixels per cell (set to (8, 8)), and cells per block (set to (2, 2)), which were selected based on standard configurations to balance computational efficiency and feature accuracy. The implementation was inspired by the original paper by Dalal and Triggs (2005) [3] and

adjusted slightly to resize bounding boxes to a fixed size (128, 128) to ensure consistent feature dimensions for comparison.

The Contrastive Language-Image Pretraining (CLIP) approach was implemented using pretrained models from the Hugging Face transformers library, based on the OpenAI CLIP paper. This involved preprocessing input images with CLIPProcessor, resizing them to different resolutions (224x224, 128x128, and 64x64) to study the effect of patch size, and extracting embeddings using three pretrained models (ViT-B/32, ViT-B/16, and ViT-L/14). Cosine similarity was used to compare the embeddings. The choice of model architectures allowed for an analysis of embedding dimensionality and model size on performance. Modifications included resizing input images to simulate varying patch sizes and customizing preprocessing for bounding box scenarios. This implementation highlights the flexibility of CLIP in capturing semantic relationships, even across varying resolutions.

For the bounding box experiment, we processes a dataset of images annotated with YOLO-formatted bounding boxes using Roboflow, where each annotation file contains normalized coordinates for object regions. For each image, absolute bounding box coordinates are calculated from normalized YOLO values using pixel-dimension scaling to enable precise region cropping. Two parallel feature extraction pipelines are implemented:
1) HOG features are computed using 8 orientation bins on grayscale images resized to 128 pixels, preserving edge and texture patterns
2) CLIP features are extracted via a pretrained Vision Transformer (ViT-B/32) model, processing RGB regions resized to 224 pixels to match its input requirements. The implementation computes feature vectors for both full images and all bounding box regions, then calculates cosine similarity scores between full-image and region-based descriptors. This design enables direct comparison of how well each method maintains feature consistency between global and local image content. The experiment iterates through all images. Results are aggregated to compare HOG's texture-based similarity patterns with CLIP's semantic-aware features, providing insights into their respective strengths for localized vs. global image analysis. The modular implementation uses OpenCV for image processing, scikit-image for HOG computation, and Hugging Face Transformers for CLIP integration, ensuring reproducibility across diverse image datasets.

6 Experimentation results

6.1 Use case 1: Sensitivity to noise in the image

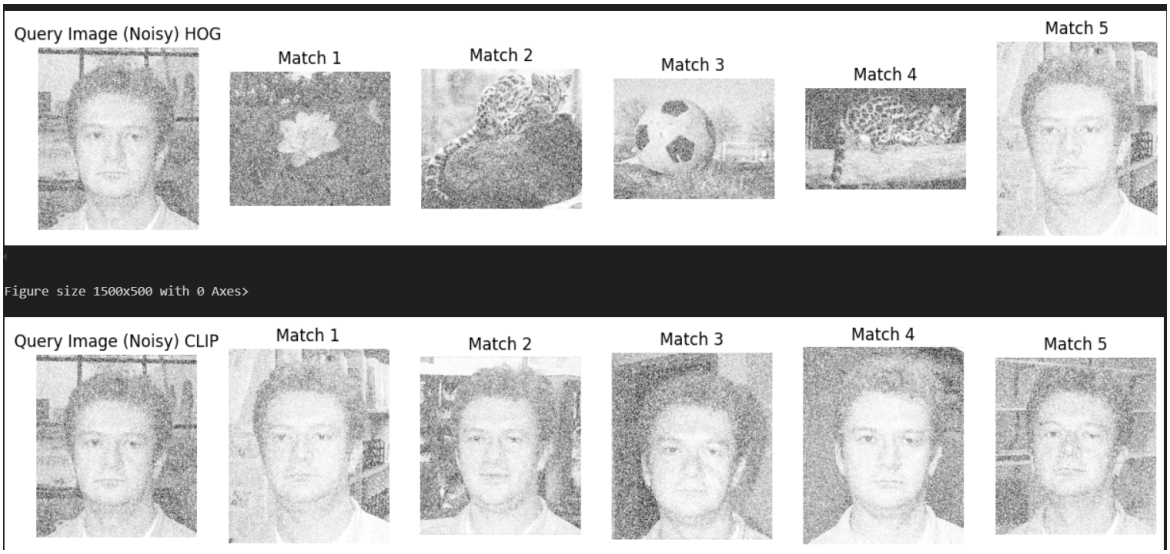

Figure 10: CLIP vs HOG in Noise sensitivity

Results demonstrate CLIPs superior performance, retrieving five semantically relevant matches (e.g., human faces) despite noise, while HOG returned only four matches with higher susceptibility to texture degradation. CLIPs resilience stems from its semantic feature space, which prioritizes high-level concepts over low-level gradients, enabling it to generalize through noise. In contrast, HOGs reliance on localized edge patterns makes it sensitive to pixel-level distortions, leading to mismatches when noise disrupts structural details.
It is worth mentioning that both models will gradually lose performance as we add the Gaussian filter, but what is guaranteed is that CLIP will always maintain higher performance than HOG, even in worse cases.

6.2 Use case 2: Sensitivity to Object Structure

The results reveal distinct matching patterns: HOG prioritizes structural/textural alignment like edges, contours, which may include objects with similar shapes or gradients but lack semantic relevance. In contrast, CLIP focuses on semantic consistency, retrieving objects that align with the query. The divergence highlights CLIPs robustness to geometric variations and HOGs sensitivity to local gradients. For tasks requiring semantic understanding, CLIP outperforms HOG, whereas HOG remains

useful for detecting structurally identical instances.

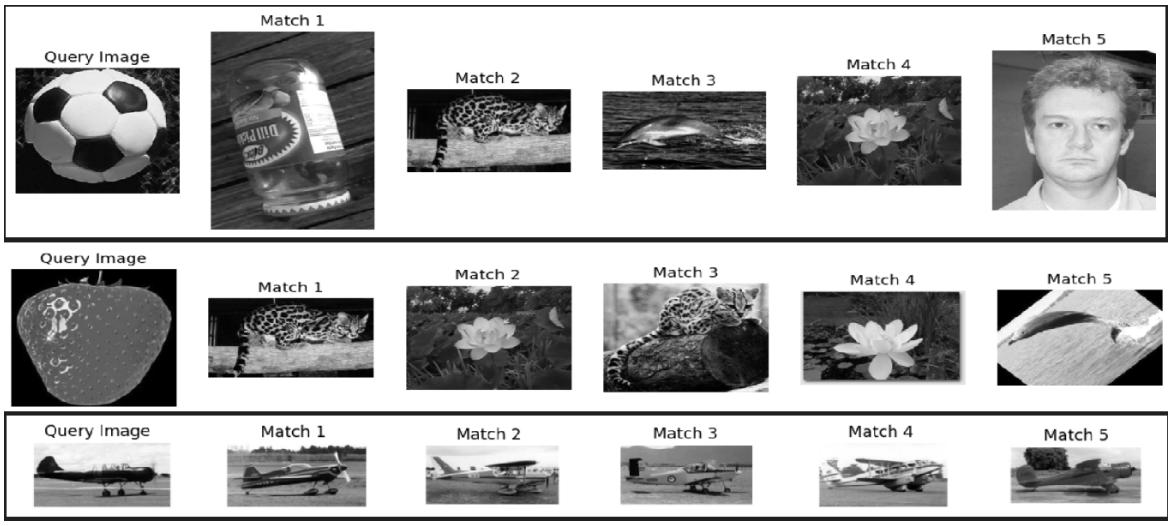

Figure 11: HOGs output to different objects according to their structure

6.3 Performance hypotheses concerning the bounding boxes

The experimental results reveal distinct performance patterns for CLIP and HOG features in relation to bounding box usage. CLIP demonstrates superior performance with bounding boxes, achieving consistently high semantic similarity scores (0.75 to 0.98) across diverse object categories. Its vision-language training enables robust recognition of localized regions, even when objects are partially visible (e.g., lotus3: 0.96 CLIP similarity despite fragmented petals) or occupy small areas of the image (e.g., cars: 0.79 CLIP similarity). This suggests CLIP benefits from bounding boxes by focusing on semantically relevant regions while ignoring distracting backgrounds. In contrast, HOG performs better with bounding boxes also, but maybe not as well as CLIP, as its reliance on full-image texture gradients and spatial layouts makes it sensitive to cropping. HOG scores drop significantly for small or fragmented regions (e.g., lotus3: 0.43 HOG similarity) and struggles with objects occupying limited areas (e.g., cars: â¤0.68 HOG similarity). While HOG excels when objects dominate the frame (e.g., cat2: 0.92 HOG similarity), its dependence on edge continuity limits its utility for localized analysis. These findings indicate CLIP is better suited for object-centric tasks requiring semantic alignment, whereas HOG remains effective only for full-image texture analysis. For applications demanding both semantic and spatial precision, a hybrid approach combining CLIPs bounding box-aware features with HOGâs full-image texture descriptors could optimize performance.

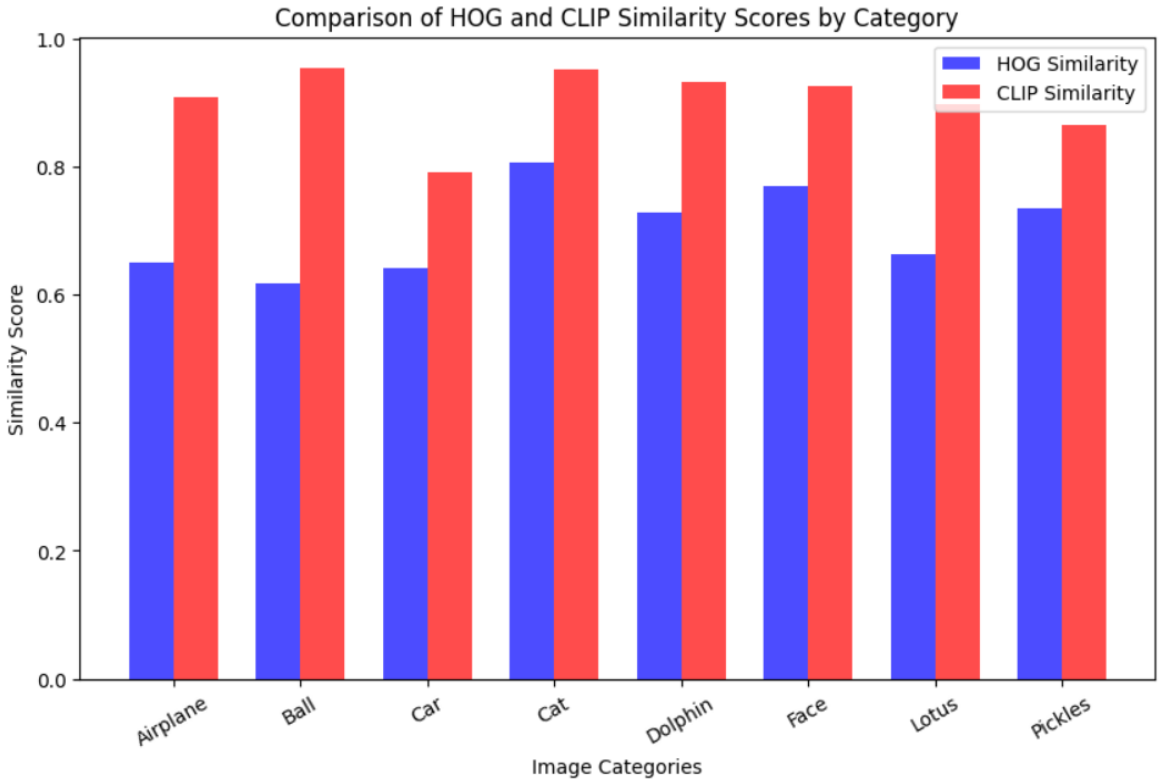

Figure 12: Comparison of HOG and CLIP similarity scores by category

7 Discussion on results and prior hypotheses

The results support the hypotheses regarding sensitivity to noise, object structure, and bounding box effects. HOG-based similarity is more affected by noise than CLIP, as evidenced by its lower and more inconsistent similarity scores, confirming that HOG relies heavily on edge-based features while CLIP maintains robust performance due to its semantic understanding. Additionally, HOG demonstrates greater sensitivity to object structure changes, with noticeable drops in similarity for objects with less defined edges, whereas CLIP remains stable. The bounding box hypothesis is also validated, as HOG similarity varies significantly with cropping and misalignment, while CLIP remains consistent. However, certain cases, such as "pickles" exhibit mixed behavior, suggesting the need for more controlled tests. To improve these evaluations, systematic noise levels, structured shape variations, and a bounding box ablation study could be implemented to better quantify their effects.

8 Conclusion

In conclusion, our analysis highlights the fundamental differences between HOG and CLIP in content-based image retrieval. HOG, being a feature-based descriptor, is highly sensitive to noise, object structure, and bounding box variations, making it less robust in uncontrolled environments. In contrast, CLIP demonstrates strong resilience to these variations due to its semantic understanding of images, leading to more consistent similarity scores. While HOG remains useful for capturing fine-grained edge details, its limitations in real-world applications are evident. The results suggest that CLIP is a more reliable choice for general image retrieval tasks, especially when robustness to variations is required. However, combining both approaches could potentially enhance retrieval accuracy by leveraging the strengths of each method. Future work should focus on refining testing methodologies, exploring hybrid models, and further analyzing the impact of different image perturbations on retrieval performance.

List of Figures

References

[1] "Learning Transferable Visual Models From Natural Language Supervision." https://arxiv.org/pdf/2103.00020v1. Accessed: 2025-01-20.

[2] "the image and text encoder in CLIP." https://www.researchgate.net/publication/380347156_Mind-bridge_reconstructing_visual_images_based_on_diffusion_model_from_human_brain_activity/figures. Accessed: 2025-01-20.

[3] "Histograms of Oriented Gradients for Human Detection." https://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf. Accessed: 2025-01-20.