Few-Shot Logo Recognition

Oggero Paolo
s342937
address:
TODO: insert email address

Cancemi Alessia
s347156
address
TODO: insert email address

Mincigrucci Gabriele
s358987
address
TODO: insert email address

Abstract

Logo recognition in open-world scenarios is hampered by the long-tailed nature of the data. This work presents a Few-Shot Learning pipeline based on LogoDet-3K and Deep Metric Learning. Using a ResNet-50 optimized with Triplet and Contrastive Loss, we map logos into a 128-dimensional embedding space. Through a progressive freezing strategy, the model learns generalized representations that allow the retrieval of brands never seen during training. The results demonstrate that optimizing the latent space geometry significantly improves the mAP and F1-score compared to standard classification methods.

1. Introduction

Accurate logo recognition has become a fundamental pillar of media analysis and copyright protection. In uncontrolled contexts, however, computer vision systems must contend with market dynamics: new brands emerge every day with unique visual identities that must be instantly identified. Classical classification paradigms suffer from two structural limitations: the need for enormous datasets for each class and the inability to recognize categories not included in the training set. These models are rigid and vulnerable to the long-tailed distribution of the real world, where rare classes are the norm. Few-Shot Learning emerges as a necessary solution, shifting the focus from ”mnemonic recognition” to ”morphological comparison.” In this work, we address this challenge by implementing a Deep Metric Learning system.

Instead of training the model to assign a label, we train it to generate embeddings in a compact latent space. Using the LogoDet-3K dataset, we developed a pipeline that extracts deep features using a ResNet-50 and projects them into a metric space where the Euclidean distance reflects the semantic relatedness of the logos. This approach not only better manages data sparsity, but also allows the system to operate on unseen classes without the need for retraining. The approach presented here focuses on overcoming the limitations of traditional classification through various technical solutions. The work first introduces an embedding-based architecture, developed using a linear projection head that enables a standard CNN to extract metric features. This structure allows for an in-depth comparative analysis of losses, evaluating how Triplet and Contrastive Loss (in the Euclidean and Cosine variants) influence the topology of the latent space. To support training, a dynamic transfer learning management system based on progressive layer unlocking is implemented, ensuring an optimal balance between model stability and learning speed, while also allowing for deeper specialization of the extracted features. The work concludes with an Open-Set Evaluation conducted on brands never encountered during the training phase, using retrieval metrics such as mAP to validate the system’s actual generalization capability.

Figure 1. Sample of images from LogoDet-3K

2. Data

For this project, we used the LogoDet-3K dataset[1], which currently represents one of the largest and most complex benchmarks for logo recognition.

2.1. Dataset Description

LogoDet-3K comprises 3,000 logo categories, with approximately 200,000 manually annotated objects distributed across 158,652 images. The dataset is hierarchically organized into nine supercategories (Food, Clothes, Necessities, Others, Electronics, Transportation, Leisure, Sports, and Medical). The dataset is inherently long-tailed, we can see that in the figure below 2, some classes have a very large number of samples, while others (especially in the Medical or Sports categories) have very few instances. This distribution faithfully reflects the challenges of real-world scenarios. In 2 the classes are grouped by macrocategory along the x-axis, and the number of images per class is shown on the y-axis. The red dots indicate the class with the most images for each macrocategory. We can see a significant imbalance between macrocategories, both in terms of the number of classes per macrocategory and the number of images per class.

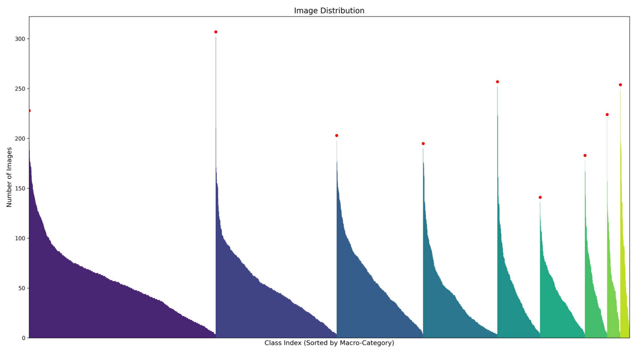

Figure 2. Distribution of the LogoDet-3K dataset by macro-category

2.2. Dataset Preprocessing and Partitioning

The preparation pipeline, aimed at optimizing dataset integrity and loading efficiency, was divided into three phases. **Brand Consolidation**: Automated cleanup was performed to merge duplicate folders (e.g., alpinestart-1 and alpinestars-2 contained the same information) and normalize labels in XML annotations. **Static Indexing**: To accelerate data loading, a unique brand-to-index mapping was generated offline, injected directly into the XML files, and saved in a CSV lookup file for quick reference. **Brand-Wise Split**: The dataset was split offline, isolating 10% of the brands for the test set. This separation at the class level (rather than at the individual image level) ensures rigorous open-set evaluation, testing the model on categories never encountered during training. At runtime, the system manages an online split of the remaining data using the getTrainValPaths function. This allows you to dynamically split the brands between Training and Validation (70/30 split for routine tests and 80/20 for final runs), ensuring that validation always occurs on classes not used for weight updating.

2.3. Sampling and Data Loading Strategies

To meet the requirements of the implemented loss functions, two specific Dataset classes were created: **DatasetTriplet**: For each reference image (Anchor), randomly selects one example from the same brand (Positive) and one from a different brand (Negative). **DatasetContrastive**: Generates image pairs with a 50% probability of belonging to the same brand (label 1) or different brands (label 0).

2.4. Preprocessing and Data Augmentation

The original images have heterogeneous resolutions. In the loading module, the data is normalized and transformed to improve the model's robustness:

- **Resize and Normalization:** All images are resized to 224×224 pixels and normalized using the ImageNet mean and standard deviation, ensuring compatibility with the pre-trained weights of ResNet-50.
- **Augmentation:** During training, transformations are applied, including `RandomResizedCrop`, `RandomHorizontalFlip`, and `ColorJitter`. These techniques simulate the distortions typical of real-world logos (light variations, angled shots, etc.).

3. Methods

This section describes the system architecture and the optimization methodologies adopted to transform the logo recognition problem into a Deep Metric Learning task.

3.1. Model Architecture: LogoResNet50

The implemented architecture is based on a ResNet-50 backbone, chosen for its optimal balance between computational depth and feature extraction capability. The original model, pre-trained on ImageNet, has been modified to adapt to the metric learning paradigm through two structural interventions:

- **Embedding Head Evolution**: Different configurations for latent space projection were tested, varying the size of the embeddings (128 and 256) and the module complexity. In addition to the single Linear layer, a more robust architecture was evaluated consisting of a first high-dimensional FC layer, followed by **1D Batch Normalization** and **Dropout**, and ending with a second FC layer for the final embedding. This structure is designed to regularize learning and improve class separation.
- **Spatial Normalization**: To ensure the effectiveness of the loss functions, the embeddings are normalized using the Euclidean norm. This step prevents the model from minimizing the loss by artificially increasing the magnitude (scale) of the vectors to distance them. Normalization forces the network to optimize the **semantic direction** of the logos, ensuring that spatial proximity reflects a true morphological similarity and not simply a scale artifact.

3.2. Loss Functions

To optimize the topology of the embedding space, three different training objectives were implemented and compared:

- **Triplet Margin Loss**: This is the primary retrieval strategy. The function works on triplets (Anchor, Positive, Negative) and minimizes the distance between Anchor and Positive, while simultaneously maximizing the distance between Anchor and Negative with a configurable margin ($\alpha = 1.0$).
- **Contrastive Loss (Euclidean)**: Optimized for image pairs, it penalizes the distance between similar pairs and forces dissimilar pairs to a distance greater than the set margin.
- **Contrastive Loss (Cosine)**: A variant of the previous one that uses cosine similarity instead of Euclidean distance. This metric is particularly effective in high-dimensional spaces because it focuses on the orientation of the vectors rather than their magnitude.

3.3. Training and Transfer Learning Strategy

Training is managed by a dynamic configuration system (Config) that allows for reproducible experiments. The main phases include:

- **Progressive Freezing**: To preserve pre-learned knowledge about ImageNet, the model starts training with frozen backbone convolutional blocks (freeze_layers).
- **Dynamic Unfreezing**: Upon reaching a predetermined epoch (unfreeze_at_epoch = 5), the code unfreezes layers, allowing for fine-tuning of logo-specific spatial features. To handle the sudden increase in trainable parameters, a learning rate reduction was implemented, allowing for more stable and precise fine-tuning of spatial features.
- **Optimization**: Both **SGD** and **Adam** were evaluated, adopting a differentiated learning rate strategy for both, where the learning rate for the backbone parameters is reduced by a factor of 1 compared to that for the head parameters. This allows us to refine the specific morphological knowledge of the logos without compromising the patterns pre-trained on ImageNet.

3.4. Evaluation Metrics

The main metrics calculated in the validation loop are:

- **F1-Score**: Dynamically calculated during training to monitor the balance between Precision and Recall.
- **Mean Average Precision (mAP)**: Used to measure retrieval quality, i.e., how effectively the model ranks correct logos at the top of search results.
- **Precision and Recall**: Calculated to analyze the model's behavior with respect to false positives and false negatives. Precision indicates how reliable the positive predictions (logo matches) are, while recall measures the system's ability to retrieve all correct instances.
- **Recall at fixed Precision (R@95P)**: It measures the percentage of correct matches recovered when the system detects a high accuracy (95%), that is, when you want to minimize false positives.

4. experiments

The experimental analysis consisted of a series of tests aimed at identifying the optimal combination of hyperparameters, following different transfer learning strategies.

4.1. Configuring Training and Testing

For training and evaluating the system, a pipeline has been defined that separates feature learning from their evaluation in unseen scenarios.

- **Training (Deep Metric Learning)**: All experiments were conducted using a ResNet-50 pre-trained on ImageNet and adapted to the metric learning paradigm via an embedding head. Training was performed using a progressive unfreezing strategy: the backbone was initially frozen to stabilize the projection in latent space, then progressively unfreezed to adapt the features to the morphology of the logos.
- **Testing (Few-Shot Evaluation)**: Testing is not based on simple class accuracy, but rather on episodic evaluation (N-way K-shot). The system is tested on brands never seen in the training set. For each episode, a few example images (support set) are provided, along with a query to be classified using cosine similarity in the latent space. This ensures that the model is effectively ”measuring similarity” and not memorizing labels.

4.2. Hyperparameter Optimization Strategy

The training phase followed an iterative optimization strategy to achieve the best hyper parameters configuration. The model was evaluated based on its ability to minimize the Triplet Loss while maximizing the Validation F1 Score at an optimal distance threshold.

4.3. Triplet Loss Analysis

The evolution of testing on Triplet Margin Loss provided critical insights into the impact of gradient stability and geometric constraints on the embedding space quality. Analysis of Model Iterations and Improvements: Preliminary experiments conducted on a 10,000-image subset revealed several technical challenges that shaped the final training: Data Augmentation: To mitigate early overfitting 3, a pipeline including perspective shifts, rotations, and color jitter was implemented. This forced the ResNet-50 backbone to move beyond simple pixel matching and learn affine-invariant features 4. Optimization and Unfreezing: Transitioning from Adam to SGD yielded more stable convergence. Furthermore, "multi-stage unfreezing" tests showed that aggressive backbone releases, where different parts of the backbone were unfrozen at different epochs, induced gradient "shocks" 5. This led to the adoption of a more conservative approach with a single unfreezing only of the later stage of the ResNet backbone. L2 Normalization and the "Scaling Problem": Initial runs without L2 normalization revealed a fundamental vulnerability: the model minimized loss by globally expanding distances (a trivial scaling strategy). This caused a persistent upward trajectory in the distance threshold as it can be seen in Figures from 3 to 5. Introducing L2 normalization and a hidden linear layer forced the model to map logos onto a unit hypersphere, favouring threshold convergence and ensuring a more robust latent space. Optimal Model Results 6: The final configuration demonstrated the robustness required for Open-Set scenarios: Loss and F1-Score Curves: Constant convergence was observed, with Validation Loss stabilizing around 0.45. The F1-score reached a peak of 0.775; the sharp improvement at epoch 15 validates the progressive unfreezing protocol combined with a learning rate reduction (factor 0.2). Threshold Stability: Unlike "unconstrained" models, the optimal distance threshold converged asymptotically toward around 1.18. This stability confirms that the model transitioned from a scaling strategy to a structured latent space with reliable boundaries, essential for identifying unseen categories.

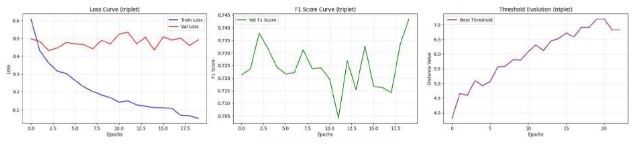

Figure 3. Early overfitting

Table 1. Iperparametri di configurazione - Test 1

Table 2. Configuration hyperparameters related to the results in Figure 3.

hyperparameter	**Value**
Batch Size	32
Margin	1.0
Learning Rate	0.001
Embedding Layer	Sequential (Linear 2048 $\rightarrow$ 128)
Freeze	5 (Fixed)

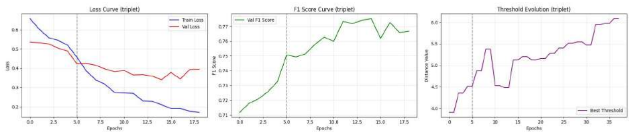

Figure 4. shifts, rotations, and color jitter implementation

Table 3. Configuration hyperparameters related to the results in Figure 4.

hyperparameter	**Value**
Dataset Size	10000
Batch Size	64
Margin	1.0
Learning Rate	$1.0 \times 10^{-3} \rightarrow 1.0 \times 10^{-4}$ (Decay factor: 0.1)
Weight Decay	True
Embedding Layer	MLP (2048 $\rightarrow$ 1024 $\rightarrow$ ReLU $\rightarrow$ 128)
Freeze Strategy	5 $\rightarrow$ 4 (Sblocco all'epoca 5)
Data Augmentation	Resize, Perspective ($p = 0.3$), H-Flip ($p = 0.3$), Rotation ($\pm 15°$), Grayscale ($p = 0.1$), ColorJitter, Normalize

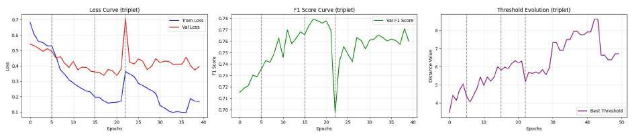

Figure 5. backbone unfreezing "shock"

Table 4. Configuration hyperparameters related to the results in Figure 5.

hyperparameter	**Value**
Optimizer	SGD (Momentum $= 0.9$)
Batch Size	64
Dataset Size	10000
Margin	1.0
Learning Rate	5.0×10^{-2} (Constant, Factor: 1)
Weight Decay	True
Embedding Layer	MLP (2048 $\rightarrow$ 1024 $\rightarrow$ ReLU $\rightarrow$ 128)
Freeze Strategy	5 $\rightarrow$ 4 (ep 5) $\rightarrow$ 5 (ep 15) $\rightarrow$ 3 (ep 22)
Data Augmentation	Resize (224×224), RandomPerspective ($p = 0.3$), RandomHorizontalFlip ($p = 0.3$), RandomRotation ($\pm 15°$), RandomGrayscale ($p = 0.1$), ColorJitter, Normalize

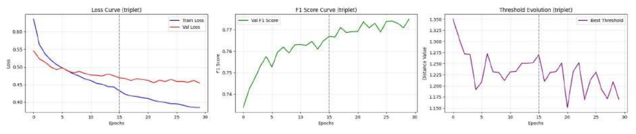

Figure 6. Optimal Model Results

Table 5. Configuration hyperparameters related to the results in Figure 6.

hyperparameter	**Value**
Dataset Size	Full
Batch Size	256
Optimizer	SGD
Weight Decay	0.0001
Margin	1.0
Learning Rate (Head)	$1.0 \times 10^{-1} \rightarrow 2.0 \times 10^{-2}$ (Decay Factor: 0.2)
Backbone LR Factor	0.03 (rispetto al LR della Head)
Embedding Layer	MLP (2048 $\rightarrow$ 256 $\rightarrow$ BatchNorm $\rightarrow$ ReLU $\rightarrow$ Dropout($p = 0.3$) $\rightarrow$ 128)
Freeze Strategy	4 $\rightarrow$ 3 (Sblocco all'epoca 15)
Data Augmentation	Resize (224×224), RandomPerspective ($p = 0.3$), RandomHorizontalFlip ($p = 0.3$), RandomRotation ($\pm 15°$), RandomGrayscale ($p = 0.1$), ColorJitter, Normalize

4.4. Contrastive Loss Analysis (Cosine)

The experimental analysis of the variant based on Cosine Similarity reveals a distinct learning dynamic compared to the triplet-based approach, focusing on the refinement of the angular margin between embeddings. Analysis of Model Iterations and Improvements: Optimizer Stability: Initial tests on a 10,000-sample subset compared the stability of SGD and Adam. While the SGD baseline showed a consistent upward trend in F1-score 7, the Adam optimizer introduced significant volatility in the distance threshold 8. This reinforced that momentum-based SGD is more effective at maintaining a rigid geometric margin for this task. Architectural Refinements: Utilizing the refinements developed during the Triplet Loss phase, specifically the inclusion of a hidden linear layer and L2 normalization, the contrastive runs focused on optimizing the relationship between positive and negative pairs. The L2 normalization is particularly critical here, as Cosine Similarity inherently operates on the unit hypersphere, ensuring that the model optimizes the angle between vectors rather than their magnitude. Full-Scale Optimization: The final scale-up to the full dataset utilized a batch size of 256 to better fit the Colab environment and a 0.25 backbone learning rate factor. Releasing part of the backbone weights after a certain epoch allowed the embedding head to stabilize before the final fine-tuning phase. Optimal Model Results: The combination of Contrastive Loss and the previously optimized hyperparameters successfully established a reliable global metric: F1-Score and Convergence: As shown in Figure 9, the validation F1-score peaked at 0.811, outperforming the triplet-based baseline. This suggests that the Contrastive approach provides a more efficient learning for the model under the selected hyperparameter configuration. By directly optimizing the similarity between pairs, the model achieved better convergence compared to the triplet-based strategy, which appears to be a more complex approach in this case. Threshold Stability: The distance threshold successfully stabilized at approximately 0.60. This steady convergence proves that the model achieved a structured latent space where the "intra-class" similarity and "inter-class" distance are well-defined.

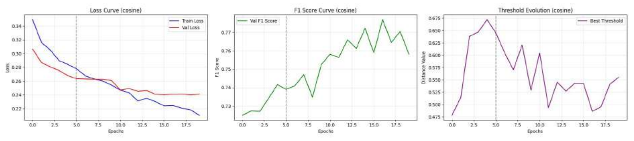

Figure 7. todo

Table 6. Configuration hyperparameters related to the results in Figure 7.

hyperparameter	Value
Dataset Size	10000
Batch Size	64
Optimizer	SGD
Weight Decay	0
Margin	0.2
Learning Rate	1.0×10^{-2} (Costante, Factor: 1)
Backbone LR	Factor 0.5 (rispetto al LR Head)
Output Norm.	True
Embedding	MLP (2048 $\rightarrow$ 1024 $\rightarrow$ ReLU $\rightarrow$ 256)
Freeze Strat.	4 $\rightarrow$ 3 (Sblocco all'epoca 5)
Data Aug.	Resize (224 $\times$ 224), RandomPerspective (p = 0.3), RandomHorizontalFlip (p = 0.3), RandomRotation ($\pm 15°$), RandomGrayscale ($p = 0.1$), ColorJitter, Normalize

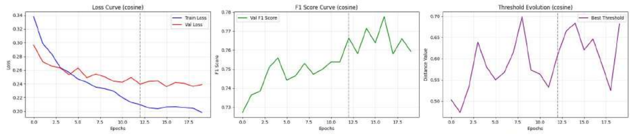

Figure 8. todo

Table 7. Configuration hyperparameters related to the results in Figure 8.

hyperparameter	Value
Dataset Size	10000
Batch Size	64
Optimizer	Adam
Weight Decay	0
Margin	0.2
Learning Rate	$1.0 \times 10^{-4} \rightarrow 1.0 \times 10^{-5}$ (Decay Factor: 0.1)
Backbone LR	Factor 0.5 (rispetto al LR Head)
Output Norm.	True
Embedding	MLP (2048 $\rightarrow$ 1024 $\rightarrow$ ReLU $\rightarrow$ 256)
Freeze Strat.	4 $\rightarrow$ 3 (Sblocco all'epoca 12)
Data Aug.	Resize (224 $\times$ 224), RandomPerspective (p = 0.3), RandomHorizontalFlip (p = 0.3), RandomRotation ($\pm 15°$), RandomGrayscale ($p = 0.1$), ColorJitter, Normalize

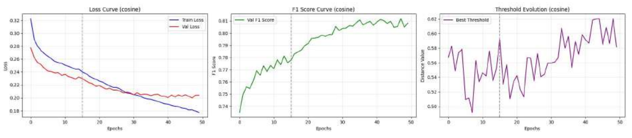

Figure 9. todo

Table 8. Configuration hyperparameters related to the results in Figure 9.

hyperparameter	Value
Dataset Size	Full
Batch Size	256
Optimizer	SGD
Weight Decay	0
Margin	0.2
Learning Rate	1.0×10^{-2} (Costante, Factor: 1)
Backbone LR	Factor 0.25
Output Norm.	True
Embedding	MLP (2048 $\rightarrow$ 1024 $\rightarrow$ ReLU $\rightarrow$ 256)
Freeze Strat.	4 $\rightarrow$ 3 (Sblocco all'epoca 15)
Data Aug.	Resize (224 $\times$ 224), RandomPerspective (p = 0.3), RandomHorizontalFlip (p = 0.3), RandomRotation ($\pm 15°$), RandomGrayscale ($p = 0.1$), ColorJitter, Normalize

4.5. Comparison between Baseline and Proposed Models

The following table presents the results obtained comparing the proposed models with the baseline.

Metric	Testing Triplet	Testing Cosine	Baseline
Accuracy	0.8602	0.8794	0.8618
Precision	0.2660	0.2787	0.2473
Recall	0.4503	0.4390	0.4003
F1-Score	0.2679	0.2769	0.2432
R@95p	0.2840	0.2999	0.2610
mAP	0.4460	0.4550	0.4176
J	0.3926	0.3801	0.3716

Table 9. Comparison between Baseline and Proposed Models.

The superiority of the proposed model stems primarily from three factors. First, domain specialization: while the ResNet-50 baseline is pre-trained on ImageNet and therefore optimized for natural objects, our model was retrained on LogoDet-3K. Thanks to progressive unfreezing, the convolutional filters adapted to the morphological characteristics of the logos, improving feature discriminability. Second, the structuring of the embedding space. In the baseline, the embeddings are not explicitly optimized to separate the classes, while the use of Triplet and Contrastive Loss forces the formation of compact clusters for the same brand and increases the distance between different classes. This is reflected in the improvement in mAP (from 0.4176 to 0.4550), a sign of more accurate and consistent ranking. Finally, an increase in retrieval capacity is observed, with an improvement in recall (from 0.4003 to 0.4725). The model therefore recovers a greater share of correct matches within the database.

4.6. Qualitative Analysis of Retrieval Behavior

Figure 1 shows a qualitative example of the retrieval system's behavior in the embedding space. The image on the left represents the query (anchor), belonging to the Count Chocula brand. In the center is a positive sample of the same brand, while on the right is a negative sample belonging to a different brand. The model assigns a similarity of 0.5291 to the positive sample, sufficient to pass the decision threshold and produce a correct match. Conversely, the negative sample achieves a significantly lower similarity (0.3422) and is correctly rejected. This behavior highlights the model's ability to place images of the same brand in adjacent regions of the latent space, while maintaining a clear separation from different brands. The most notable aspect of this example is not the absolute value of the similarity, but the stable margin between positive and negative. The distance between the two scores is sufficient to ensure a robust decision even in the presence of visual variations or samples belonging to unseen classes. In an open-set retrieval system, reliability depends on how consistently the model maintains this separation margin over time. The figure also highlights how the model doesn't limit itself to recognizing superficial elements of the logo, but is able to capture morphological features shared between objects of the same brand even when the visual appearance varies significantly (e.g., packaging vs. character figurines). This suggests that the learned embedding space does not simply encode textures or colors, but a more abstract semantic representation of the brand's visual identity. This qualitative example supports the quantitative results obtained in the experiments: the stability of the decision threshold observed during training translates into consistent system behavior even in real-world retrieval cases.

4.7. Qualitative Embedding Analysis

Visualization of the latent space using t-SNE qualitatively confirms the quantitative results. The points corresponding to the different brands form distinct and relatively compact clusters, with a clear separation between different categories. This behavior suggests that the model does not simply memorize surface patterns, but learns morphological

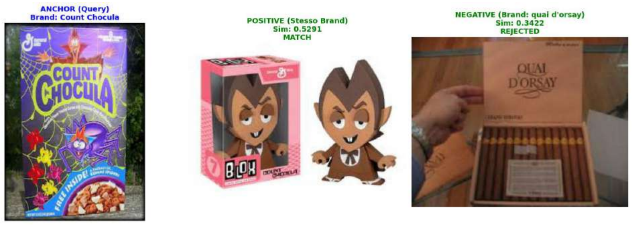

Figure 10. Qualitative analysis of retrieval behavior (Anchor, Positive, Negative).

features shared between logos of the same brand.

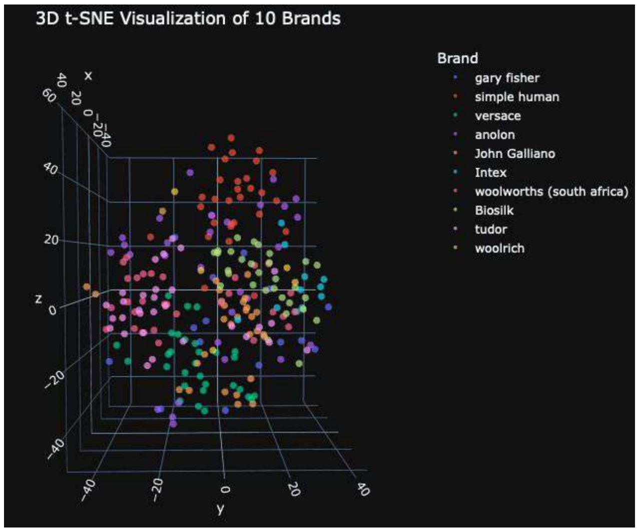

Figure 11. 3D t-SNE Visualization of 10 Brands.

5. Conclusion

This project demonstrated that logo recognition in open-set scenarios can be successfully addressed through Deep Metric Learning. The adoption of an embedding-based paradigm allowed us to overcome the limitations of closed classification systems, enabling the retrieval of brands never seen in the training phase. Experimental analysis highlighted a key result: the quality of a retrieval system is not measured solely by peak accuracy, but by the stability of the separation between classes in the latent space. A robust model is one capable of maintaining a consistent decision threshold over time, avoiding oscillations that could compromise the system's reliability. In this context, Triplet Loss has been shown to produce a more stable embedding space geometry, making it more suitable for industrial applications such as copyright protection and automatic media analysis.

5.1. Problems encountered

During training and testing, we were severely limited by the computational resources available to us; for this reason, we partially ran the model locally on our devices with a limited number of samples. Additionally, we also used Colab's A100 units, mitigating but not completely solving the problem. By limiting ourselves to a portion of the LogoDet-3K dataset, we don't know how the learning would have performed on the entire dataset.

Overfitting: The most prevalent problem during the training phases was the strong tendency towards overfitting; this was the focus of hyperparameter improvement. We used data augmentation to make the model more resilient to brand distortions. Of all the data augmentation techniques, we avoided image cropping; this makes the task much more complex. At the same time, we reduced the backbone learning rate to avoid feature degradation, thus avoiding a loss of backbone generalization and recall of key features. We also used the dropout technique to develop a more robust model, reducing error variance. Seeking to further improve the model's performance, we applied normalization to each branch; this, not reflecting normalization across the entire dataset, introduces statistical noise from the mean and variance approximations computed on the mini-batch. The effect on training performance is similar to dropout, promoting faster convergence.

5.2. future developments

Better results can likely be achieved by training the network on triplets, using hard negative mining or hard example mining; this would theoretically improve robustness by selecting the most informative triplets for training, i.e., those in which the "positive" image is distant from the anchor and the "negative" image is close to the anchor, forcing the network to correct the error. This would also lead to a more discriminative embedding space. For the sake of completeness, it's worth mentioning that other types of losses and architectures could be explored for this type of project; one example is Faster R-CNN. Region proposal could be better at mitigating the problem of logos presented in natural, complex, or cluttered environments, and could yield good results when used as a backbone. As mentioned, different losses could also be used, including a variation of the triplet loss in which multiple negatives are proposed simultaneously; this could accelerate convergence by providing more separation constraints between negatives and anchors/positives. Another loss that could be experimented with is Proxy-Anchor Loss, which teaches the network a proxy vector for each class; these proxies are compared to each image, and through training, each image is pushed toward one of the proxies, theoretically allowing for greater separation between class clusters.

References

[1] Jing Wang, Weiqing Min, Sujuan Hou, Shengnan Ma, Yuanjie Zheng, and Shuqiang Jiang. Logodet-3k: A large-scale image dataset for logo detection. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 18(1s):1–23, 2022.