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 an embedding space. Through a progressive freezing strategy, the model learns generalized representations that allows the retrieval of brands never seen during training.

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. 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.

Figure 1: Sample of images from LogoDet-3K

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 distance reflects the similarity 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 work 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.

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. This work is based on the use of the LogoDet-3K dataset, publicly accessible via the repository: `https://github.com/Wangjing1551/LogoDet-3K-Dataset`.

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.

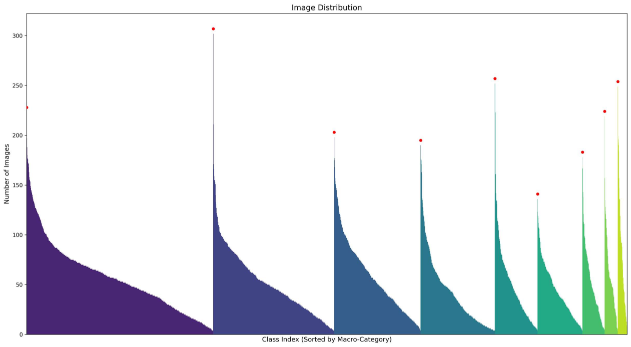

Figure 2: Distribution of the LogoDet-3K dataset by macrocategory

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, randomly selected from each category to guarantee the same data distribution. 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. we share our pre-processed dataset at: `https://drive.google.com/file/d/1yFiHK_r6ae8Xs8uls8F-ccXHPpqVyh74/view?usp=sharing`.

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. This process involves **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. Subsequently, **Augmentation** is applied during training, transformations, 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 a custom **Embedding Head**. This module consists of a high-dimensional FC layer, followed by *1D Batch Normalization* and *Dropout*, ending with a second FC layer for the final latent projection. To ensure the effectiveness of the loss functions, the embeddings undergo **Spatial Normalization** using the Euclidean norm. This step forces the network to optimize the **semantic direction** of the logos, ensuring that spatial proximity reflects true **morphological similarity** rather than scale artifacts.

3.2. Loss Functions

To optimize the topology of the embedding space, three different training objectives were implemented and compared. The first implemented method is *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 ($\alpha = 1.0$). Alternatively, the model uses *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. Finally, the third objective is *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. The process initiates with *Progressive Freezing* To preserve pre-learned knowledge about ImageNet, the model starts training with frozen backbone convolutional blocks (freeze_layers). This phase is followed by *Dynamic Unfreezing*: upon reaching a predetermined epoch we tested unfreezeing some layers, allowing for fine-tuning of logo-specific spatial features. To handle the sudden increase in trainable parameters we experimented with a learning rate reduction, allowing for more stable and precise fine-tuning. Finally, regarding *Optimization*: Both **SGD** and **Adam** were evaluated, adopting in later runs a differentiated learning rate strategy for both, where the learning rate for the backbone parameters is reduced by a config factor 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.
- **discriminant ratio (J)**: It is calculated as the ratio between the average distance of negative pairs and the average distance of positive pairs. A higher J indicates a better separation between classes in the embedding space, which is crucial for effective retrieval.

Table 1: Range of hyperparameters (Minimum - Maximum) observed across all experimental configurations.

Hyperparameter	**Min Value**	**Max Value**
Batch Size	32	256
Learning Rate (Initial)	1.0×10^{-4}	1.0×10^{-1}
Margin	0.2	1.0
Weight Decay	0	1.0×10^{-4}
Backbone LR Reduction Factor	0.03	0.5
Embedding Output Dim.	128	256

4. experiments

The experimental analysis consisted of a series of tests aimed at identifying the optimal combination of hyperparameters, following different transfer learning strategies. For clarity, the ranges of the hyperparameters used are shown in 1.

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. **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.

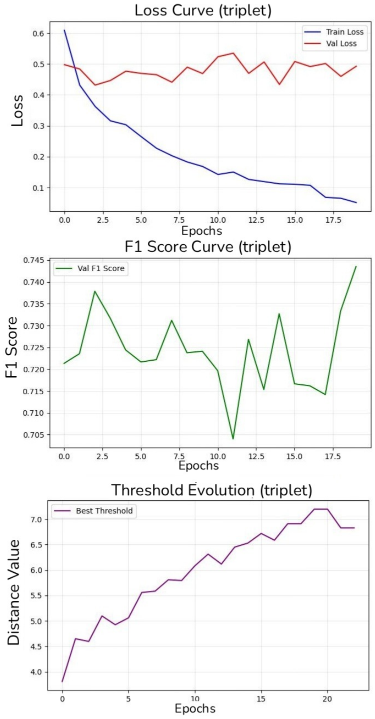

Figure 3: Early overfitting trend observed during preliminary experiments without augmentation.

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

Hyperparameter	Value
Dataset Size	10,000
Batch Size	32
Optimizer	Adam
Weight Decay	None
Margin	1.0
Learning Rate	0.001
Backbone LR Red.	None
Output Norm.	No
Embedding Layer	MLP (2048 → 128)
Freeze Strat.	None
Data Aug.	None

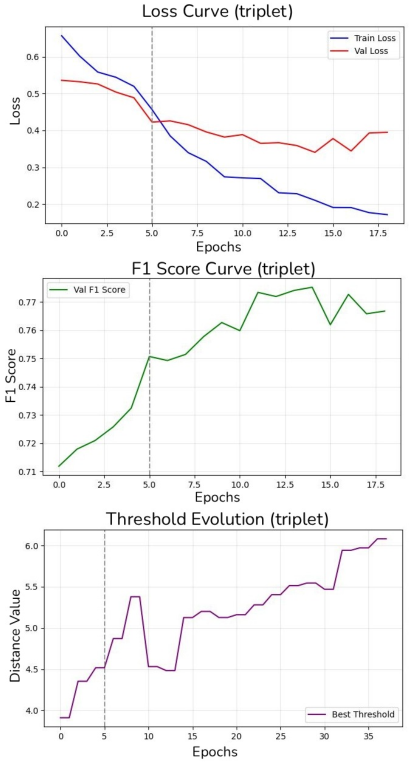

Figure 4: shifts, rotations, and color jitter implementation

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. Initially, regarding **Data Augmentation**, to mitigate early overfitting (Figure 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** (Figure 4).

Subsequently, focusing on **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” (Figure 5). This led to the adoption of a more conservative approach with a single unfreezing only of the later stage of the ResNet backbone.

Finally, to address **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 (Figure 3) to (Figure 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 (Figure 6), demonstrated the robustness required for Open-Set scenarios. Specifically, the **Loss and F1-Score Curves:** Constant convergence was

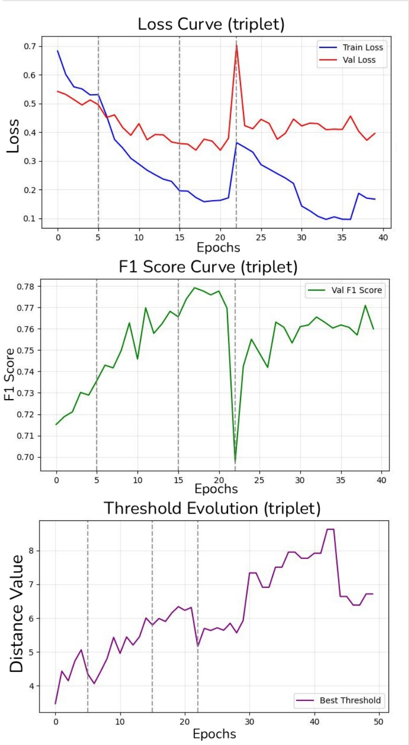

Figure 5: backbone unfreezing ”shock”

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). Moreover, regarding **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.

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:* The tuning of the hyper parameters for the contrastive loss was done taking into account the results obtained from the triplet runs. Initially, regarding *Optimizer Selection* testing on a 10,000-sample subset identified SGD as the more stable choice for this architecture. While Adam introduced significant volatility in the distance threshold during early trials, the SGD baseline (Figure 7) demonstrates a consistent, stable upward trend in F1-score and a more stable distance threshold. This stability suggests that momentum-based SGD is better suited for main-

Table 3: Configuration hyperparameters related to the results in Figure 6.

hyperparameter	**Value**
Dataset Size	Full
Batch Size	256
Optimizer	SGD (Momentum $= 0.9$)
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 (compared to the Head’s LR)
Embedding Layer	MLP (2048 $\rightarrow$ 256 $\rightarrow$ BatchNorm1d $\rightarrow$ 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

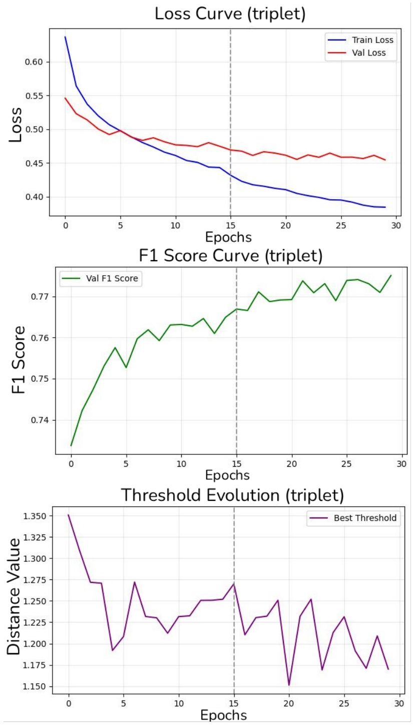

Figure 6: Optimal Model Results

taining the rigid geometric margin required for this metric learning task. Furthermore, concerning *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. Finally, the *Full-Scale Optimization* on the complete 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 (Figure 8): The combination of *Contrastive Loss* and the previously optimized hyperparameters successfully established a reliable global metric. *F1-Score and Convergence:* As visible in the graphs for the optima model, 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. *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.

4.5. Comparison between Baseline and Proposed Models

The following table presents the results obtained comparing the proposed models with the baseline across different support set sizes (n).

Comparison with Baseline: While the ResNet-50 baseline achieves high Accuracy, it struggles with retrieval-specific metrics. Our proposed models demonstrate a substantial leap in *Recall*, *mAP* and the *discriminant ratio (J)* indicating that the Metric Learning approach is more effective at clustering similar logos than standard classification. The most striking difference is the *discriminant ratio (J)*, which rises from 0.3898 in the baseline to nearly 2.0, reflecting a more optimized and expansive latent space. *Few-Shot Support ($n = 5$):* All models exhibit a performance gain as n increases. By averaging the embeddings of five support samples, the system creates a stable brand prototype, filtering out "contextual noise" (unrelated background elements). This is most visible in the Contrastive model's Recall, which jumps from 0.4732 to 0.6441. *Comparison between Models Performance:* The Contrastive Cosine approach yields superior results compared to the Triplet model. This disparity is likely because the Triplet task is inherently more difficult to optimize; without the im-

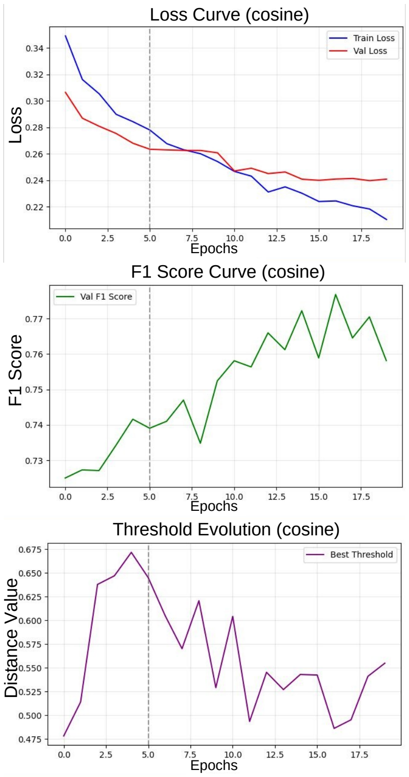

Figure 7: SGD baseline

Table 4: 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 (224×224), RandomPerspective ($p = 0.3$), RandomHorizontalFlip ($p = 0.3$), RandomRotation ($\pm 15°$), RandomGrayscale ($p = 0.1$), ColorJitter, Normalize

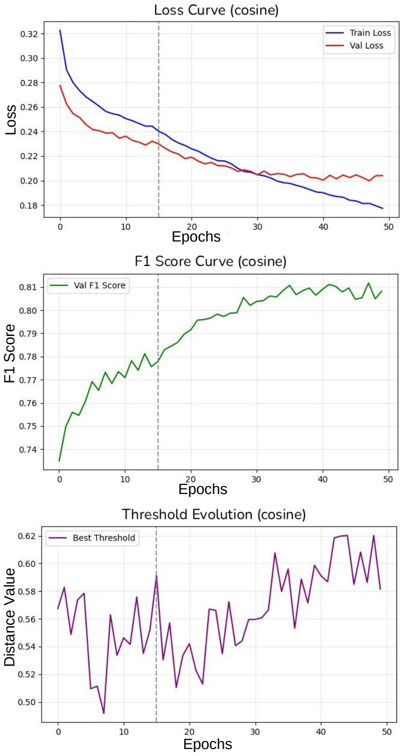

Figure 8: Optimal Model Results

1-Shot Evaluation ($n = 1$)

Metric	Triplet (Thr: 0.52)	Contrastive (Thr: 0.52)	Baseline (Thr: 0.90)
Accuracy	0.9214	0.9227	0.9185
Precision	0.3254	0.3453	0.3035
Recall	0.4525	0.4732	0.3295
F1-Score	0.3392	0.3523	0.2651
R@95p	0.3035	0.3486	0.3114
mAP	0.5083	0.5512	0.4703
J	1.9311	1.9868	0.3898

5-Shot Evaluation ($n = 5$)

Metric	Triplet (Thr: 0.50)	Contrastive (Thr: 0.56)	Baseline (Thr: 0.75)
Accuracy	0.9298	0.9209	0.9420
Precision	0.3774	0.4065	0.4072
Recall	0.5223	0.6441	0.3493
F1-Score	0.3940	0.4458	0.3350
R@95p	0.3696	0.4481	0.4506
mAP	0.5881	0.6519	0.6015
J	1.9311	1.9868	0.3898

Table 5: Comparison between Baseline and Proposed Models for 1-Shot and 5-Shot evaluation.

plementation of hard negative mining to select challenging samples, the Triplet loss struggles to refine decision boundaries as effectively as the Contrastive loss. *Threshold Optimization and Class Imbalance:* The optimal similarity thresholds (τ) were determined by maximizing the F1-score across a range of distance candidates. A notable discrepancy exists between the thresholds identified during validation ($\tau \approx 1.175$ for Triplet; $\tau \approx 0.59$ for Contrastive) and those utilized in the final evaluation ($\tau \approx 0.50$–0.56). This shift is primarily driven by the transition from a balanced validation environment to a retrieval-based testing scenario. While validation optimizes for a 1:1 positive-to-negative ratio, the testing phase reflects a realistic distribution where negative distractors vastly outnumber positive matches. In this imbalanced setting, maintaining a high F1-score requires a more stringent threshold to prevent the massive pool of negatives from generating enough False Positives to overwhelm the metric. **Analysis of Metric Variations and Model Impact:** Regarding *Classification Metrics (Accuracy, Precision, Recall, F1-Score)*, The Baseline achieves the highest Accuracy, particularly at $n = 5$, while both metric-learning models show a slight decrease, indicating a small trade-off in overall classification correctness. While the precision is not significantly impacted, the most significant improvement is observed in Recall, which increases substantially in both Triplet and Contrastive models, with the latter reaching 0.6441 at $n = 5$ compared to 0.3493 for the Baseline, as a result, the F1-Score improves consistently. Overall, these variations indicate that metric learning enhances retrieval effectiveness despite a marginal reduction in Accuracy. Concerning *R@95p* the Contrastive model improves R@95p at $n = 1$ and remains very close to the Baseline at $n = 5$, while the Triplet model shows slightly lower values. This suggests that the Contrastive loss better preserves high-confidence retrieval performance, maintaining strong recall under strict precision constraints. Furthermore,*mAP* improves noticeably with metric learning, particularly for the Contrastive model, which achieves the best overall performance at both $n = 1$ and $n = 5$. Since mAP reflects ranking quality across thresholds, this confirms that the learned embeddings produce a more consistent and discriminative similarity ordering. Finally, the *discriminant ratio (J)* increases dramatically from 0.3898 in the Baseline to nearly 2.0 in both proposed models. This substantial rise indicates a significantly more structured and separable latent space, demonstrating improved intra-class compactness and inter-class dispersion regardless of support size.

4.6. Qualitative Analysis of Retrieval Behavior

The Contrastive Loss-based architecture proves to be the most effective of the project, achieving a mAP of **65.2%**, demonstrating an abstraction capability that goes beyond simple pixel-matching. Cases like *Maille* highlight how the network extracts invariant features, correctly associating logos on complex backgrounds (e.g., cans on a black background). However, the addition of massive graphic elements, as in the case of *Captain Morgan*, can push the embedding beyond the tolerance threshold, masking the brand's identity and leading to false negatives.

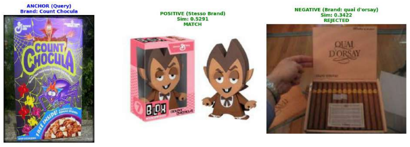

Figure 9: Qualitative analysis of retrieval behavior (Anchor, Positive, Negative).

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 features shared between logos of the same brand.

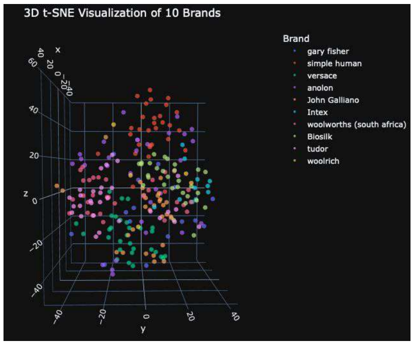

Figure 10: 3D t-SNE Visualization of 10 Brands.

4.8. Future Developments

A key limitation of this work was hardware availability. Early experiments were conducted locally on a reduced dataset due to computational constraints, and although final training runs used Google Colab A100 instances, broader access to dedicated high-performance hardware would have enabled more extensive experimentation and improved performance.

Future improvements could focus on refining the sampling strategy. The current Triplet Loss approach could benefit from *Hard Negative Mining*, encouraging the model to learn finer discriminative features by selecting more challenging positive and negative samples. Additionally, alternative losses, such as Proxy-Anchor Loss, potentially leading to improved performance.

Another challenge was processing uncropped images, which introduced significant contextual noise. Incorporating an object detection stage (e.g., Faster R-CNN) to isolate logos could allow the backbone to focus on brand-specific morphology and improve accuracy. Additionally, exploring other kinds or architectures such as Transformers may further enhance class separation and overall performance.

5. Conclusion

This project served as a comprehensive bridge between the theoretical frameworks discussed in class and the practical challenges of modern computer vision. Throughout the development process, we faced significant constraints that mirrored real-world AI development, such as hardware constrain, difficulty of the task and complex hyper parameters tuning. We put into practice several strategies explored during the course, including data augmentation, dropout, batch normalization and differential learning rates, to mitigate the constant risk of overfitting. The experience of training on uncropped images also provided a valuable lesson in the impact of contextual noise. We learned that without a localized region of interest, the model is forced to interpret the entire scene, which adds a layer of complexity. Ultimately, we are satisfied with having successfully applied these advanced techniques to a real-world scenario. This project allowed us to move beyond the theory of metric learning to build a compact and functional system, proving that even with limited resources, it is possible to translate classroom concepts into a functioning retrieval pipeline.

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.