Gradient Boosting Similarity in Entity Matching

Notebook for the LifeCLEF 2025 Lab at CLEF 2025

Sergei Fedorchenko[1,*,†], Sergei Arefiev[2,†]

[1]University of Zurich, Binzmühlestrasse 14 8050 Zürich Switzerland
[2]St Petersburg University, 7-9 Universitetskaya Embankment, St Petersburg, Russia, 199034

Abstract

We present a solution to the CLEF 2025 [1] animal species identification task, leveraging pretrained embedding models and a boosting classifier for pairwise similarity learning. Our pipeline combines pretrained feature extraction, pairwise embedding comparison, and supervised boosting to determine species-level similarity between image pairs. As a team "Tim Riggins" we have achieved competitive performance, obtaining scores of 0.618 target metric for our selected submission and 0.629 target metric for our best submission on the CLEF private leaderboard.

Keywords

Entity matching, wildlife, gradient boosting, images similarity

1. Introduction

Accurate recognition of individual animals from images plays a key role in ecological research, population tracking, and conservation efforts. The CLEF 2025 animal identification challenge [2] addresses this problem by providing a dataset aimed at identifying exact individuals across diverse environmental conditions and viewpoints. This task presents significant challenges due to limited labeled examples per individual, intra-species similarity, and natural variability in appearance. We propose a hybrid approach that leverages pretrained visual matchers and a boosting-based binary classifier to predict whether two images represent the same animal. This combination allows us to integrate robust visual features with supervised decision boundaries for improved entity-level recognition.

2. Related Work

2.1. Metric Learning

Metric learning aims to project images into an embedding space where semantically similar items are close and dissimilar items are far apart. Common approaches include **triplet loss** [3], which optimizes relative distances between anchor, positive, and negative samples, and **contrastive loss** [4], which operates on pairs of examples. These methods often require careful sampling or mining strategies to be effective. **ArcFace** [5] improves stability and discriminative power by adding an angular margin to the softmax loss, making it particularly effective for tasks with large numbers of classes.

2.2. Boosting Methods

Gradient boosting methods are widely used for structured and tabular data due to their strong performance, robustness to overfitting, and ability to handle heterogeneous feature types. These models iteratively build an ensemble of weak learners, typically decision trees, to minimize a loss function through stage-wise additive modeling. Among popular implementations, **LightGBM** [6] offers efficient

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain
*Corresponding author.
†These authors contributed equally.
sergei.a.fedorchenko@gmail.com (S. Fedorchenko); arefiev.mc@gmail.com (S. Arefiev)

training with support for large-scale datasets, categorical features, and custom objective functions. In the context of similarity learning, boosting models can be trained to predict whether a given pair of examples belongs to the same class by using handcrafted features derived from image embeddings.

2.3. Open Set Recognition

Open Set Recognition (OSR) aims to address the realistic scenario where a model may encounter inputs from classes not seen during training. Unlike traditional closed-set classifiers, OSR models must detect and reject unfamiliar samples rather than misclassify them. Techniques such as **OpenMax** [7] extend softmax classifiers by fitting Weibull distributions to activation vectors, enabling detection of out-of-distribution inputs. Other methods apply thresholding in embedding space using distances to class centroids or Mahalanobis scoring [8], making them compatible with metric learning approaches.

2.4. Energy-Based Models

Energy-Based Models (EBMs) provide a principled framework for modeling uncertainty and detecting out-of-distribution inputs. Instead of outputting class probabilities directly, these models compute an energy score that reflects the compatibility of input and model. In the OSR context, inputs with high energy are flagged as unfamiliar or anomalous. **Liu et al.** [9] propose training classifiers to minimize energy on in-distribution samples while maximizing it on out-of-distribution data using Outlier Exposure. EBMs can be seamlessly integrated with pre-trained embedding networks and have shown superior calibration and robustness compared to traditional softmax classifiers.

3. Dataset

The CLEF 2025 dataset includes images of individual animals across several species, such as lynxes, salamanders, and sea turtles with the distribution shown in the Table 1, supported along with rich metadata (e.g., species ID, individual ID, timestamp, and orientation) [10]. The primary task is to determine whether two images depict the same animal, making it a fine-grained entity recognition problem. Each species presents unique challenges due to differences in visual variability, camera conditions, and availability of labeled examples. The dataset also includes both known and unknown individuals, requiring models to generalize beyond the training set.

Table 1
Species population in train and test datasets

Species	Train Population	Test Population
Lynx	2,957	946
Salamander	1,388	689
Loggerhead Turtle	8,729	500

4. Task Description

The goal of the entity matching task is to determine whether a query animal image belongs to a known individual from a reference database or represents a previously unseen individual. The challenge spans multiple species and requires models to handle significant visual variability while generalizing to novel instances. Evaluation is based on two core metrics [10]: **BAKS** (Balanced Accuracy on Known Samples), which measures class-balanced accuracy over known individuals, and **BAUS** (Balanced Accuracy on Unknown Samples), which assesses performance on identifying novel individuals. The final score is computed as the geometric mean of BAKS and BAUS, encouraging balanced performance between identification and novelty detection.

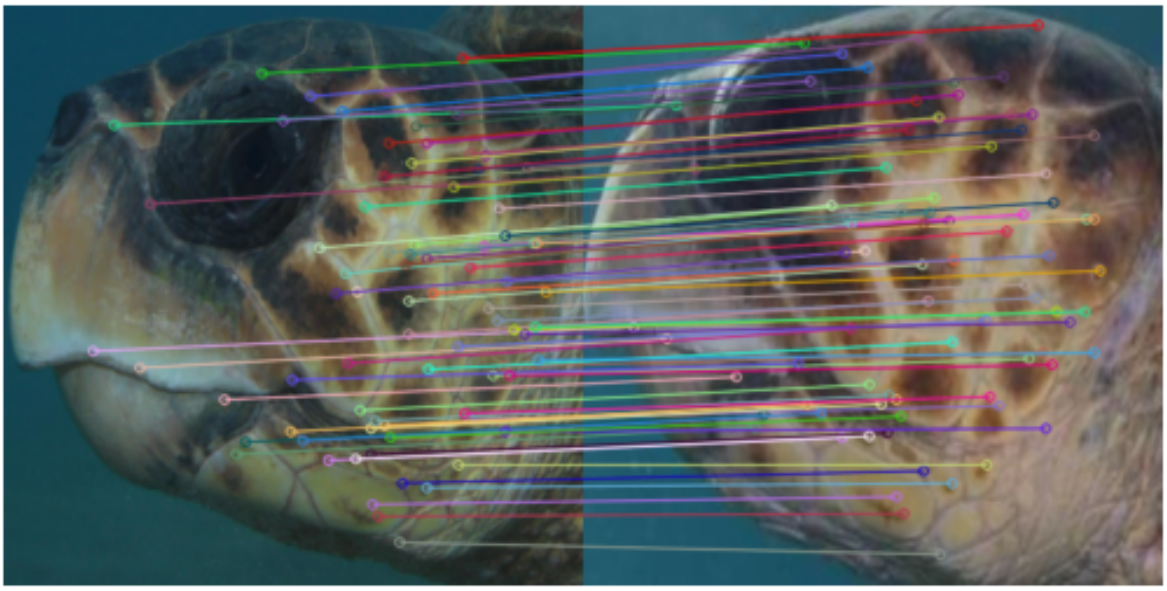

Figure 1: Automatically extracted keypoints and their matches between turtle images. Image provided by the competition hosts [11].

5. Method

5.1. Image Preprocessing

To preserve the aspect ratio of the original images, we applied padding before resizing. Images were then resized to 384×384 pixels when using the MegaDescriptor model, and to 512×512 pixels for all other local descriptor extractors. This preprocessing ensured consistency across the input pipeline while maintaining the structural integrity of key visual features.

For normalization, we used ImageNet statistics (mean and standard deviation) when working with MegaDescriptor. For the other models, min-max normalization.

5.2. Global Feature Extraction

MegaDescriptor [12] was used as our primary model for global feature extraction. Although originally designed for local descriptor learning, MegaDescriptor can also produce dense feature maps that serve as strong global representations when properly combined. We extracted fixed-length embeddings for each image without any additional fine-tuning, using these as input for direct similarity comparisons. The embeddings were used as the foundation for cosine similarity baselines. This approach allowed us to leverage powerful pre-trained representations while maintaining a training-free feature construction pipeline.

5.3. Local Feature-Based Matching

We experimented with local descriptor extractors to match animal instances based on fine-grained visual details. Specifically, we used three pre-trained models: **ALIKED** [13], **Disk** [14], and **SuperPoint** [15], all designed for keypoint detection and local descriptor extraction. Each model extracts sets of keypoints and associated descriptors that can be matched across images to establish local visual correspondences. These matches serve as a complementary signal to global embeddings, especially in cases where texture, pose, or background differences make purely global similarity less reliable.

5.4. Keypoints Information Aggregation

To combine keypoint information from multiple local descriptor extractors, we used **LightGlue** [16] as a lightweight and flexible matching framework. LightGlue was applied to the outputs of **ALIKED**, **Disk**, and **SuperPoint**, performing keypoint matching between image pairs for each method independently. The resulting correspondences were then aggregated to produce a richer and more reliable representation of local visual similarity. This aggregation step allowed us to leverage complementary strengths of different detectors—such as scale invariance, robustness to viewpoint changes, and localization precision—to improve the quality of our pairwise features. The combined matches were used as part of the input to our boosting model for final prediction.

5.5. Pairwise Feature Construction

For each image pair, we compute cosine similarity and the number of corresponding points from local feature extractors Table 2 ending up with 9 features per sample pair. These serve as input features for the boost classifier.

Table 2
Descriptions of input features used for pairwise image similarity classification.

Orientation Metadata
First image orientation
Second image orientation
Global Descriptor Similarity
MegaDescriptor similarity
Number of SuperPoint Keypoint Matches
SuperPoint (score > 0.5)
SuperPoint (score > 0.8)
Number of DISK Keypoint Matches
DISK (score > 0.5)
DISK (score > 0.8)
Number of ALIKED GLUE Keypoint Matches
A-Liked (score > 0.5)
A-Liked (score > 0.8)

To improve the robustness of local feature-based similarity, we applied a top-k filtering strategy using the MegaDescriptor model. For each pair of images, we extracted local descriptors and retained only K the most similar keypoint correspondences based on the distance of the descriptor. This step helps to reduce the noise from irrelevant or weak matches and emphasizes the most confident local alignments between images.

Utilizing cross-validation scheme, a smaller value of $K = 40$ was sufficient for lynxes and turtles, while salamanders required a higher threshold of $K = 150$ to maintain performance, likely due to greater visual similarity and more challenging matching conditions within that class.

5.6. Boosting Classifier

We train a LightGBM binary classifier to predict whether a given pair of images represents the same entities. Positive pairs consist of the same entities; negative pairs consist of different entities.

5.7. Validation

To better simulate the conditions of the test set, we leveraged the timestamp information available in the Kaggle dataset to construct train–test splits that follow the natural chronological order of image collection.

Table 3
LightGBM parameter configuration used for pairwise classification.

Parameter	Value
num_trees	50
num_leaves	32
learning_rate	0.1
boost	gbdt
metric	auc
objective	binary

Figure 2: Overview of the Prediction and Training Workflow per Pair.

In addition, we held out 20% of the individuals during training and used them as unseen entities. This setup allowed us to evaluate both identification of known individuals and detection of new ones, closely aligning with the evaluation protocol of the challenge.

Based on our validation results, we selected a threshold of 0.65 to distinguish new individuals from known ones.

5.8. Inference

During inference, we followed the same procedure as in training. For each query image, we identified its K most similar reference images based on embedding similarity. These top-K pairs were then passed through the boosting model to obtain binary match scores. The final prediction was assigned based on the reference image with the highest predicted similarity score among the K candidates. Approach depicted in the Figure 2 allowed us to efficiently combine retrieval-based filtering with supervised matching for robust entity recognition.

6. Experiments and Results

Table 4
Ablation results comparing different combinations of global features, local features, and meta-model types. Scores are reported for private and validation splits.

Global Features	Local Features	Meta-model	TH	Private Score	Validation Score	Padding
Yes	Yes	Boosting	0.75	**0.629***	0.681	Yes
Yes	Yes	Boosting	0.6	0.618	**0.686**	Yes
Yes	Yes	Average	0.6	0.608	0.669	Yes
Yes	Yes	Average	0.6	0.597	0.652	No
Yes	No	Boosting	0.6	0.322	0.341	No
Yes	No	Average	0.6	0.308	0.313	No

We report performance in the Tables 4, 5 using the official CLEF evaluation metrics. Our model outperforms simple cosine thresholding and benefits from the added discrimination of the supervised boosting model. Both the use of boosting instead of averaging, and the application of padding, lead to improved results.

Table 5
Performance on different datasets at threshold $TH = 0.75$.

Dataset	Score @ TH = 0.75
Lynx	0.7470
Salamander	0.5769
SeaTurtle	0.7272

Despite the overall consistency between validation and private leaderboard scores, some discrepancies were observed. For instance, the configuration with a threshold of 0.75 achieved the highest private score (0.629), while a lower threshold of 0.6 yielded the best validation score (0.686). This suggests that our validation split, although timeline-based and stratified, may not perfectly reflect the distribution or difficulty of the private test set. Overfitting to the validation threshold likely accounts for this performance gap, highlighting the importance of evaluating across multiple thresholds and data partitions. In future work, incorporating more robust cross-validation or ensembling across decision thresholds could help bridge this mismatch.

Finetuning MegaDescriptor within a multi-class classification framework led to noticeable performance improvements over using frozen embeddings. The model was trained to predict individual entities directly, which encouraged more discriminative representations. However, this approach proved computationally expensive and required extensive training data for convergence. In contrast, keypoint-based matching methods offered a more efficient alternative, especially when used with pretrained extractors and lightweight post-processing. As a result, we prioritized methods that combined pretrained descriptors with retrieval and pairwise scoring.

This pipeline demonstrated strong generalization performance and remained stable across validation and test scenarios. By combining global and local features with a lightweight boosting classifier, it effectively captured both coarse and fine-grained visual similarities. The method required minimal finetuning and was computationally efficient at inference time. As a result, it secured 7th place on the private leaderboard of the CLEF 2025 animal re-identification challenge.

6.1. Ablation Study

We evaluated several baseline strategies to understand the contribution of each component in our pipeline. A simple cosine similarity between pretrained embeddings yielded limited performance, especially in distinguishing hard negative pairs. We also experimented with training classification models and using their penultimate-layer embeddings for cosine-based retrieval; while this improved over the vanilla approach, it still lacked robustness. In contrast, our boosting-based method, which leverages pairwise features, consistently outperformed both baselines by capturing more nuanced relationships between image pairs. This highlights the value of supervised modeling on relational features beyond raw embedding distances.

7. Discussion

The pretrained embedding model captured useful semantic structure. Boosting provided flexible decision boundaries in embedding space. Triplet-based training was unstable in our setup, and the pretrained matcher consistently outperformed self-trained models.

8. Conclusion and Future Work

We presented a simple yet effective framework for entity matching based on pretrained visual embeddings and a supervised boosting model trained on pairwise features. Rather than relying solely on embedding distance thresholds, our approach learns to model similarity through handcrafted relational features and a gradient boosting classifier, providing more flexibility and robustness to noise or domain

shifts. This framework proved particularly useful in scenarios where the embedding space alone was insufficient to capture fine-grained species-level distinctions. In future work, we plan to explore self-supervised pretraining to improve the embedding quality, incorporate graph-based label propagation to exploit the structure of embedding similarity graphs, and investigate hierarchical clustering techniques to capture taxonomic relationships between species.

9. Declaration on Generative AI

During the preparation of this work, the author(s) used ChatGPT in order to: Grammar and spelling check. The changes were minimal and affected only grammar and punctuation, not the meaning of the text. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

Acknowledgments

We thank the CLEF 2025 organizers and baseline authors for providing pretrained models and evaluation scripts.

References

[1] L. Picek, S. Kahl, H. Goëau, L. Adam, T. Larcher, C. Leblanc, M. Servajean, K. Janoušková, J. Matas, V. Čermák, K. Papafitsoros, R. Planqué, W.-P. Vellinga, H. Klinck, T. Denton, J. S. Cañas, G. Martellucci, F. Vinatier, P. Bonnet, A. Joly, Overview of lifeclef 2025: Challenges on species presence prediction and identification, and individual animal identification, in: International Conference of the Cross-Language Evaluation Forum for European Languages (CLEF), Springer, 2025.

[2] L. Adam, K. Papafitsoros, R. Kovář, V. Čermák, L. Picek, Overview of AnimalCLEF 2025: Recognizing individual animals in images, 2025.

[3] E. Hoffer, N. Ailon, Deep metric learning using triplet network, 2018. URL: https://arxiv.org/abs/1412.6622. arXiv:1412.6622.

[4] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, 2021. URL: https://arxiv.org/abs/2004.11362. arXiv:2004.11362.

[5] J. Deng, J. Guo, J. Yang, N. Xue, I. Kotsia, S. Zafeiriou, Arcface: Additive angular margin loss for deep face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (2022) 5962–5979. URL: http://dx.doi.org/10.1109/TPAMI.2021.3087709. doi:10.1109/tpami.2021.3087709.

[6] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, 2017.

[7] A. Bendale, T. Boult, Towards open set deep networks, 2015. URL: https://arxiv.org/abs/1511.06233. arXiv:1511.06233.

[8] K. Lee, K. Lee, H. Lee, J. Shin, A simple unified framework for detecting out-of-distribution samples and adversarial attacks, 2018. URL: https://arxiv.org/abs/1807.03888. arXiv:1807.03888.

[9] W. Liu, X. Wang, J. D. Owens, Y. Li, Energy-based out-of-distribution detection, 2021. URL: https://arxiv.org/abs/2010.03759. arXiv:2010.03759.

[10] L. Adam, V. Čermák, K. Papafitsoros, L. Picek, Wildlifereid-10k: Wildlife re-identification dataset with 10k individual animals, 2025. URL: https://arxiv.org/abs/2406.09211. arXiv:2406.09211.

[11] L. Adam, V. Čermák, K. Papafitsoros, L. Picek, Seaturtleid2022: A long-span dataset for reliable sea turtle re-identification, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 7146–7156.

[12] V. Čermák, L. Picek, L. Adam, K. Papafitsoros, Wildlifedatasets: An open-source toolkit for animal re-identification, 2023. URL: https://arxiv.org/abs/2311.09118. arXiv:2311.09118.

[13] X. Zhao, X. Wu, W. Chen, P. C. Y. Chen, Q. Xu, Z. Li, Aliked: A lighter keypoint and descriptor extraction network via deformable transformation, 2023. URL: https://arxiv.org/abs/2304.03608. arXiv:2304.03608.
[14] M. J. Tyszkiewicz, P. Fua, E. Trulls, Disk: Learning local features with policy gradient, 2020. URL: https://arxiv.org/abs/2006.13566. arXiv:2006.13566.
[15] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, 2018. URL: https://arxiv.org/abs/1712.07629. arXiv:1712.07629.
[16] P. Lindenberger, P.-E. Sarlin, M. Pollefeys, Lightglue: Local feature matching at light speed, 2023. URL: https://arxiv.org/abs/2306.13643. arXiv:2306.13643.

A. Online Resources

All code can be found at the github repository https://github.com/SergeyFedorchenko/AnimalCLEF25_7th.git.