Individual Wildlife Recognition via Hybrid Global-Local Matching and Segmentation-Aware Filtering

Notebook for the LifeCLEF Lab at CLEF 2025

Roman Pakhomov[1,†], Grigory Demidov[1,†], Kristian Bogdan[1,†], Svyatoslav Lanskikh[2,*,†], Danis Dinmuhametov[1,†] and Andrey Khlopotnykh[1,†]

[1]National Research University Higher School of Economics (HSE University), 11 Pokrovksy Bulvar, Moscow, 109028, Russian Federation

[2]Central University (CU University), 7 Gasheka Street, Moscow, 123056, Russian Federation

Abstract

Animal re-identification allows for non-invasive, scalable monitoring of wildlife by matching visual cues specific to each individual. In this paper, we present a mapping-oriented, open system that progressively filters and validates candidate images. We took the basic step-by-step solution presented in the competition and improved each step in it. First, we calculate global attachments to select a small subset of potentially suitable images for each query. Then we use a set of local feature matching tools, each of which contains a separate detector—descriptor pair and a matching algorithm, to obtain additional similarity estimates that capture the smallest visual details, such as unique markings and morphological patterns. Then we combine these estimates for each user using a studied, weighted synthesis mechanism that identifies the most reliable features for different types and shooting conditions. Finally, a calibrated confidence threshold allows you to separate previously seen individuals from new ones, ensuring reliable recognition when new animals are detected. We are evaluating based on the AnimalCLEF 2025 collections (loggerhead sea turtles, salamanders, and Eurasian lynx), and our system provides highly balanced accuracy for both known and unknown classes. The modular design makes it easy to adapt additional mapping devices or embedded models, demonstrating resistance to background interference, blockages, and variable shooting conditions. With this solution, we took the first place in the AnimalCLEF2025 competition.

Keywords

Animal Re-identification, Open-Set Identification, Wildlife Conservation, Computer Vision, LifeCLEF 2025

1. Introduction

Animal re-identification is a critical task in wildlife research and conservation, enabling the tracking of individuals over time to study population dynamics, habitat use, migration patterns, and behavior. By recognizing unique traits—such as markings, color patterns, or morphological features—researchers can monitor species in a non-invasive, scalable manner. Automating this process not only accelerates data collection but also enhances the consistency and scale at which individuals can be reliably tracked. These capabilities are vital for identifying biodiversity threats and supporting evidence-based conservation strategies.

Despite recent progress in computer vision and machine learning, reliably identifying individual animals remains challenging. Models often overfit to environmental cues such as background, lighting, or camera angle—rather than focusing on species-specific, individual characteristics. This results in poor generalization to new environments or image conditions, limiting the practical effectiveness of many re-identification systems in real-world conservation settings.

The AnimalCLEF 2025 challenge, part of the LifeCLEF 2025 evaluation campaign, addresses this problem through the task of individual animal identification for three wildlife species: loggerhead sea turtles (*Caretta caretta* [1]) sourced from Zakynthos, Greece; salamanders (*Salamandra salamandra*) from the Czech Republic; and Eurasian lynxes (*Lynx lynx*) also from the Czech Republic. For each test image, the objective is to determine whether the animal has been seen before (i.e., is present

CLEF 2025 Working Notes, 9 – 12 September 2025, Madrid, Spain

*Corresponding author.

†These authors contributed equally.

in the reference database) or is a new, previously unseen individual. If known, the correct identity must be assigned. To aid generalization, participants are allowed to augment their models using the WildlifeReID-10k dataset—a large-scale benchmark comprising over 140,000 images across 10,000+ individuals from a diverse set of species [2, 3].

In this technical report, we present our solution to the AnimalCLEF 2025 challenge. Our pipeline consists of four core stages: global candidate selection, local visual matching, score aggregation (bagging), and novelty filtering. This modular structure combines coarse-to-fine similarity evaluation with a confidence-based thresholding mechanism to distinguish between known and novel individuals. Our approach balances precision and generalization while remaining robust across the three target species.

2. Related Work

Several prior works address fine-grained wildlife classification and open-set identification. CNN ensembles and metadata fusion prove effective for discriminating visually similar species. Local feature matchers (e.g., SuperPoint[4], DISK, ALIKED[5] + LightGlue[6]) combined with global descriptors yield robust re-identification under varying viewpoints and background clutter. Calibration methods like isotonic regression help distinguish known from unseen individuals. Our pipeline integrates these ideas to handle both coarse filtering and fine-grained matching in an open-set setting.

3. Evaluation

3.1. Evaluation Metrics

To properly evaluate the performance of our re-identification pipeline on both known and novel individuals, we employ three metrics: Balanced Accuracy on Known Samples (BAKS), Balanced Accuracy on Unknown Samples (BAUS), and their Geometric Mean (GeoMean).

3.1.1. Balanced Accuracy on Known Samples (BAKS)

The *Balanced Accuracy on Known Samples (BAKS)* quantifies the model's performance on individuals that are present in the training dataset. Unlike standard accuracy, BAKS is computed in a class-balanced manner to mitigate the effects of class imbalance.

Let C be the set of known classes. Then:

$$\text{BAKS} = \frac{1}{|C|} \sum_{c \in C} \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c},$$

where:

nosep TP_c is the number of true positive predictions for class c,
nosep FN_c is the number of false negatives for class c.

3.1.2. Balanced Accuracy on Unknown Samples (BAUS)

The *Balanced Accuracy on Unknown Samples (BAUS)* measures the model's ability to correctly recognize individuals from *unseen* classes, i.e., classes not present in the training dataset.

Let U be the set of unknown classes. Then:

$$\text{BAUS} = \frac{1}{|U|} \sum_{u \in U} \frac{\text{TP}_u}{\text{TP}_u + \text{FN}_u},$$

where:

nosep TP_u is the number of true positive predictions for unknown class u,
nosep FN_u is the number of false negatives for class u.

3.1.3. Overall Metric: Geometric Mean (GeoMean)

To combine performance on both known and unknown samples, we compute the geometric mean of BAKS and BAUS:

$$\text{GeoMean} = \sqrt{\text{BAKS} \times \text{BAUS}}.$$

4. Approach

4.1. Data Preprocessing and Segmentation Strategy

Effective individual identification in wildlife datasets is often hindered by visual noise, background clutter, and inconsistencies in object localization. These issues are particularly prominent in the AnimalCLEF2025 dataset, where species such as salamanders are often partially occluded by human hands, and segmentation annotations are absent for several species (notably salamanders and sea turtles). To address this, we implement a targeted data preprocessing strategy centered on segmentation-aware data augmentation. For sea turtles, for instance, the ocean background in most images naturally isolates the subject, effectively acting as an implicit segmentation mask. Lynxes are already segmented.

4.1.1. Segmentation Model Fine-Tuning

We construct a supplementary annotated dataset[7] of 817 salamander images to compensate for the lack of precise segmentation masks. This dataset includes pixel-level annotations of the target individuals and is used to fine-tune a YOLOv11m-seg model[8] —an instance-aware object segmentation network with a modern convolutional backbone.

The fine-tuned YOLOv11m-seg model significantly improves the localization quality of salamander regions by learning to suppress irrelevant background areas, particularly human hands and other visual artifacts. Qualitative inspection confirms more compact and accurate segmentation masks, which are then used to crop or mask the input images during downstream embedding extraction.

Table 1
Segmentation Metrics

Metric	Value
mAP50	0.95
mAP50–90	0.90
Recall	0.91
Precision	0.95

4.1.2. Impact on Re-identification Performance

The implemented segmentation strategy significantly reduces the negative impact of background noise in species where it is dominant. Most notably, the use of segmentation for salamanders yields a measurable improvement in downstream matching metrics, confirming the hypothesis that accurate localization of the individual is a crucial prerequisite for reliable embedding computation and similarity-based identification.

4.2. Baseline Description

Our baseline system integrates global deep descriptors and local keypoint-based matching to address open-set individual identification. The implementation relies on the `wildlife-datasets` and `wildlife-tools` libraries, designed to facilitate data loading, feature extraction, and similarity computation for wildlife-related tasks. Throughout this work, the term *baseline* refers to the publicly available solution Baseline with WildFusion.

4.2.1. Pipeline Overview

The pipeline consists of the following components: Our pipeline extracts global descriptors using the MegaDescriptor-L-384[9] model from the timm library[10], performs local keypoint detection and matching with ALIKED features and MatchLightGlue, fuses the resulting global and local similarity scores via the WildFusion module, and finally calibrates the combined scores using isotonic regression.

4.2.2. Global Descriptor Extraction

Global features are extracted from a high-capacity vision transformer model `MegaDescriptor-L-384`, pre-trained and fine-tuned to generate robust embeddings. These embeddings capture coarse-grained appearance information and are compared using cosine similarity.

Given two embeddings $\mathbf{e}_x, \mathbf{e}_y \in \mathbb{R}^d$, cosine similarity is defined as:

$$\mathrm{sim}_{\cos}(\mathbf{e}_x, \mathbf{e}_y) = \frac{\mathbf{e}_x \cdot \mathbf{e}_y}{\|\mathbf{e}_x\| \, \|\mathbf{e}_y\|}.$$

4.2.3. Local Feature Extraction and Matching

The baseline uses `ALIKED` as the local keypoint detector and descriptor. It is designed for accurate and efficient keypoint extraction with descriptor computation. The extractor outputs dense descriptors $\mathbf{d}_i$ and confidence scores.

These local descriptors are matched using `MatchLightGlue`, an attention-based matcher inspired by `LightGlue`. Let α_{ij} denote the attention affinity between descriptor i in image x and j in image y:

$$\alpha_{ij} = \frac{\exp(\mathbf{W}_q \mathbf{d}_i^x \cdot \mathbf{W}_k \mathbf{d}_j^y)}{\sum_{j'} \exp(\mathbf{W}_q \mathbf{d}_i^x \cdot \mathbf{W}_k \mathbf{d}_{j'}^y)}.$$

After filtering matches via mutual nearest neighbors, the local similarity score $s_\ell(x, y)$ is:

$$s_\ell(x, y) = \frac{1}{|\mathcal{M}(x, y)|} \sum_{(i,j) \in \mathcal{M}(x,y)} \alpha_{ij}.$$

4.2.4. WildFusion Fusion Strategy

The `WildFusion`[11] module computes the final similarity as the average of global and local scores.

4.2.5. Score Calibration

Fused similarity scores are passed through isotonic regression, fitting a monotonic mapping g to minimize squared error between calibrated scores and ground truth labels.

4.3. Disadvantages

The basic level has a number of limitations. Initially, the approach presents only one matching method, the embedding model may not suit all species equally, the similarity matrices are averaged, and the lack of initial segmentation of salamanders creates background noise.

5. Methodology

The proposed solution encompasses a four-stage process:

Table 2
Pipeline Stages

Stage	Approach	Description
1	Initial Candidate Selection	Identify top-B global embedding matches.
2	Local Feature Matching	Compare candidates using multiple extractor–matcher pairs.
3	Weighted Aggregation (Bagging)	Combine per-pair scores via learned weights $w_{e,m}$.
4	Novelty Detection	Apply threshold τ to calibrated scores to flag new individuals.

5.1. Selection of Most Relevant Instances

For each test sample x, we compute its embedding $\mathbf{e}_x \in \mathbb{R}^d$ using the pre-trained embedding model. Denote the set of all training embeddings as $\{\mathbf{e}_i^{\text{train}}\}_{i=1}^N$. We compute cosine distances:

$$d_{\cos}\big(\mathbf{e}_x,\, \mathbf{e}_i^{\text{train}}\big) \;=\; 1 \;-\; \frac{\mathbf{e}_x \cdot \mathbf{e}_i^{\text{train}}}{\|\mathbf{e}_x\|\;\|\mathbf{e}_i^{\text{train}}\|}.$$

We then select the top-B training samples with the smallest distances:

$$\mathcal{C}(x) \;=\; \arg\min_{B}\; \text{top}\Big\{\, d_{\cos}(\mathbf{e}_x,\, \mathbf{e}_i^{\text{train}})\Big\}_{i=1}^{N}.$$

These B candidates form the candidate set for local matching.

We have replaced `MegaDescriptor-L-384` with a higher-quality `miewid-msv3` [12].

5.2. Local Feature Matching

Each test image x is compared against each candidate $c \in \mathcal{C}(x)$ using several "extractor + matcher" pairs. Let

$$\mathcal{E} = \{\text{KeyNetAffNetHardNet},\ \text{DISK (Kornia[13])},\ \text{SuperPoint},\ \text{ALIKED},\ \text{DISK (wildlife\_tools)}\}$$

be the set of feature extractors, and

$$\mathcal{M} = \{\text{AdaLAM},\ \text{GADMatcher},\ \text{MatchLightGlue}\}$$

be the set of matchers. For each extractor $e \in \mathcal{E}$ and matcher $m \in \mathcal{M}$, we compute a local similarity matrix:

$$S_{e,m}(x, c) \;=\; \text{LocalSim}_{e,m}(x, c),$$

where $\text{LocalSim}_{e,m}$ denotes the local matching procedure with extractor e and matcher m. Each entry of the local similarity matrix corresponds to a confidence-weighted sum of matched keypoints.

The selection of the Kornia library was motivated by its provision of models capable of handling affine transformations, which expanded the comparison capabilities beyond the single matcher (MatchLightGlue) and limited set of extractors (SIFT, SuperPoint, ALIKED, DISK) available in `wildlife_tools`.

5.3. Bagging (Aggregation)

We replaced the inefficient averaging of the proximity matrix with a weighted vote.

Let $S_{e,m}(x, c)$ be the similarity score for a given pair (x, c) obtained from the local similarity matrix corresponding to each (e, m).

We introduce a set of coefficients, $w_{e,m}$, where each coefficient is associated with a unique combination of e and m. These coefficients allow us to assign different weights to the similarity scores from each local matrix. The final similarity score $s(x, c)$ is then computed as a weighted mean of these per-pair scores:

$$s(x,c) = \frac{1}{|\mathcal{E}| \cdot |\mathcal{M}|} \sum_{(e,m)\in\mathcal{E}\times\mathcal{M}} w_{e,m} S_{e,m}(x,c).$$

The optimal values for these coefficients, $w_{e,m}$, are determined through an optimization process. This optimization is performed on a training dataset using `scipy.optimize.minimize`[14] with the **Powell** method. The objective of this optimization is to minimize the **geomin metric** as defined by Baks and Baus, ensuring that the final similarity score $s(x,c)$ is well-calibrated for the given task.

5.4. Novelty Detection

Based on the aggregated similarity $s_{\text{agg}}(x,c)$, the most probable entity (predicted individual) and its corresponding confidence score are identified for each test image. To ascertain new individuals (i.e., those not present in the training dataset), a confidence threshold(τ) is established.

$$p_{\max}(x) \;=\; \max_{c\in\mathcal{C}(x)} p(x,c),$$

and assign:

$$\hat{y}(x) = \begin{cases} \arg\max\limits_{c\in\mathcal{C}(x)} p(x,c), & \text{if } p_{\max}(x) \;\geq\; \tau, \\ \text{new\_individual}, & \text{otherwise.} \end{cases}$$

The threshold τ is chosen to maximize $\mathrm{GeoMean}$ on the validation set:

$$\tau^* \;=\; \arg\max_{\tau\,\in\,[0,1]} \sqrt{\mathrm{BAKS}(\tau) \;\times\; \mathrm{BAUS}(\tau)}.$$

6. Results

Table 3
The results of each improvement ($B = 50$)

Score	Approach
0.369	Baseline embedding only
0.412	replacing the embedder with miewid-msv3
0.624	+ Local matching
0.639	+ Segmentation
0.648	+ Embedder in local matchers
0.668	+ Optimized fusion weights

Table 4
Final thresholds and GeoMeans per Species

Species	τ^*	GeoMean
Eurasian Lynx	0.119	0.795
Salamander	0.054	0.735
Loggerhead Sea Turtle	0.080	0.882

In the final solution, we replaced the basic embedder with a higher-quality miewid-msv3, increased the number of local matches using the kornia library, segmented the salamanders, added embedder to the local matches and made a weighted vote instead of averaging Table 3.

We selected B and add high-point-limit local matchers. Final GeoMean=0.713, yielding first place by 0.04 margin over second [15].

7. Discussion

We explore alternative embeddings (CLIP[16], DinoV2[17]) and dense matchers (LoFTR[18]). While these show promise, our species-specific pipeline remains most consistent. Future work includes species-routing hybrid matchers for further gains.

8. Acknowledgments

We thank the LifeCLEF organizers for the AnimalCLEF 2025 datasets and the WildlifeReID-10k auxiliary data. We also acknowledge developers of `wildlife-datasets` and `wildlife-tools`.

9. Declaration on Generative AI

In preparing this work, the authors used GPT-4o to check grammar and spelling. In addition, the authors use GPT-4o to translate text into English. After using these tools/services, the authors reviewed and edited the content as needed and are solely responsible for the content of the publication.

References

[1] 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.

[2] V. Čermák, L. Picek, L. Adam, K. Papafitsoros, WildlifeDatasets: An Open-Source Toolkit for Animal Re-Identification, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2024, pp. 5953–5963.

[3] L. Adam, V. Čermák, K. Papafitsoros, L. Picek, Wildlifereid-10k: Wildlife re-identification dataset with 10k individual animals, arXiv preprint arXiv:2406.09211 (2024).

[4] D. DeTone, T. Malisiewicz, A. Rabinovich, Superpoint: Self-supervised interest point detection and description, in: CVPR Deep Learning for Visual SLAM Workshop, 2018. URL: http://arxiv.org/abs/1712.07629.

[5] 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, IEEE Transactions on Instrumentation Measurement 72 (2023) 1–16. URL: https://arxiv.org/pdf/2304.03608.pdf. doi:10.1109/TIM.2023.3271000.

[6] P. Lindenberger, P.-E. Sarlin, M. Pollefeys, LightGlue: Local Feature Matching at Light Speed, in: ICCV, 2023.

[7] GG, remove background dataset, 2025. URL: https://universe.roboflow.com/gg-gcd3a/remove-background-nvy0p-kyxht, visited on 2025-06-05.

[8] G. Jocher, J. Qiu, Ultralytics yolo11, https://github.com/ultralytics/ultralytics, 2024.

[9] V. Čermák, L. Picek, L. Adam, K. Papafitsoros, Wildlifedatasets: An open-source toolkit for animal re-identification, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 5953–5963.

[10] R. Wightman, Pytorch image models, https://github.com/rwightman/pytorch-image-models, 2019. doi:10.5281/zenodo.4414861.

[11] V. Cermak, L. Picek, L. Adam, L. Neumann, J. Matas, Wildfusion: Individual animal identification with calibrated similarity fusion, arXiv preprint arXiv:2408.12934 (2024).

[12] C. X. Labs, miewid-msv3: Multi-species vertebrate identification model, https://huggingface.co/conservationxlabs/miewid-msv3, 2024.

[13] E. Riba, D. Mishkin, D. Ponsa, E. Rublee, G. Bradski, Kornia: An open source differentiable computer vision library for pytorch, in: Winter Conference on Applications of Computer Vision, 2020.

[14] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, SciPy 1.0 Contributors, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272. doi:10.1038/s41592-019-0686-2.

[15] The solution is available online at https://github.com/XXXM1R0XXX/AnimalCLEF2025, ????

[16] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, L. Schmidt, Openclip, 2021. URL: https://doi.org/10.5281/zenodo.5143773. doi:10.5281/zenodo.5143773, if you use this software, please cite it as below.

[17] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, P. Bojanowski, Dinov2: Learning robust visual features without supervision, 2023.

[18] J. Sun, Z. Shen, Y. Wang, H. Bao, X. Zhou, LoFTR: Detector-free local feature matching with transformers, CVPR (2021).