A Reproducible Resource-Aware Cascade for Wildlife Image Analysis Under Limited Data

Xixian Wang[1, a]

[1]The Ohio State University, Ohio, USA

[a]wang.16722@osu.edu

Abstract

Selecting when to invoke a costly classifier is harder than evaluating the cost-performance curve after the fact. We examine this problem in a reproducible wildlife image-analysis prototype that uses a lightweight model for routine screening and a stronger model for selected cases. The evaluation uses 1,000 verified ENA24 images from four balanced classes. Numeric-ID burst proxies and identical 64-bit difference hashes produce 160 conservative groups. Five outer group-stratified folds keep fitting, model selection, calibration, routing validation, and testing separate. Frozen ImageNet-pretrained MobileNetV3-Small and ResNet-18 extract features for multinomial logistic-regression heads. Temperature scaling is fitted on calibration data. Each fold then selects the lowest-cost threshold whose routing-validation Macro-F1 is within 0.02 of stage 2. The pooled out-of-fold Macro-F1 is 0.89535 for stage 1 and 0.92397 for stage 2. The validation-selected cascade calls stage 2 for 3.9% of test images and reaches 0.89720, a 0.02677 gap from stage 2 that exceeds the specified tolerance. Higher fixed budgets approach stage-2 performance, but they do not change the failure of the prospective selection rule. The result points to threshold-transfer instability in this small-data setting. Acoustic or other contextual signals are a possible extension, not an experiment reported here. The resulting artifact is a reproducible visual prototype with an explicit evaluation protocol and a clearly reported negative result.

Keywords

Camera traps; wildlife image classification; conditional inference; model cascade; confidence calibration; group-stratified cross-validation; resource-aware inference.

1. Introduction

Camera traps generate large image collections, making automated review useful for researchers, protected-area staff, and non-expert users. Public resources such as Snapshot Serengeti and ENA24 support wildlife computer-vision research at different scales and with different label structures [1], [2]. Deep models have automated animal identification and ecological description [3], while camera-trap software has addressed human, animal, and empty-frame filtering [4], [5]. The practical question is not only whether a model is accurate, but how much computation should be spent on each image.

A confidence-routed cascade provides one way to allocate this computation. A relatively inexpensive first model handles routine cases, while a stronger model verifies uncertain images. This is related to early-exit and adaptive-network research [6], [7], [8]. In an application, however, the threshold must be selected without using final test outcomes. Calibration can make confidence more meaningful [9], [10], although small validation sets may still produce unstable thresholds.

The data also contain a dependence problem. Nearby camera-trap frames may share backgrounds, poses, and capture context, so a random image split can overstate generalization.

Recognition can decline at unseen locations [11], and transferability remains a concern in wildlife machine learning [12]. When sequence and camera-location identifiers are unavailable, conservative proxy groups can reduce obvious leakage, but they cannot establish ecological independence.

Our evaluation involves a visual two-stage cascade based on 1,000 ENA24 images of American Black Bear, American Crow, Dog, and Virginia Opossum. The first stage uses MobileNetV3-Small and the second stage uses ResNet-18 [13], [14]. Our methodology provides a calibrated prediction at the first stage, requests better verification when required, and logs the confidence, model used, and cost estimate. Our key question is whether the pre-specified validation criterion chooses a cascade within 0.02 Macro-F1 of stage 2 on out-of-sample folds. This does not happen. New multimodal datasets and cross-modal validation techniques hint that acoustic activity may aid routing in future work [15], [16]. However, we restrict our assessment to the visual stream. Our work offers an auditable process involving burst grouping, nested partitioning, calibration, and selection of the routing. We also present a user-oriented research prototype, without using a computer-only benchmark as proof of sensor node deployment.

2. Related Work

2.1. Camera-trap analysis and generalization

The Snapshot Serengeti and ENA24 datasets serve as examples of the two typical types of camera trap data: the first one being a large annotated dataset, while the second one includes a small set of images with their respective detection metadata [1], [2]. Previous research dealt with such topics as sequence level classification, filtering of animals from empty frames, and available wildlife models [17], [4], [5], [12]. At the same time, it brings uncertainty into question, as the users should understand which predictions can be accepted as true, and which should be verified. Our task involves only four way full-frame classification for a 1000 image balanced subset.

2.2. Conditional computation and multimodal monitoring

BranchyNet, adaptive neural networks, and SkipNet perform input-dependent computation [6], [7], [8]. Our cascade is simpler. We always run a full MobileNetV3-Small extractor, and sometimes a full ResNet-18 extractor. Temperature scaling allows for confidence-based routing, while ECE and NLL are merely descriptive metrics and do not offer guarantees of reliability [9], [10]. SmartWilds brings together camera trap, acoustic, drone data, and deployment metadata [15]. Cross-Modal Corroboration verifies visual and acoustic activity measurements against behavior priors [16]. Both studies could inspire either acoustic-assisted routing or disagreement alerts, though we don't reproduce them here. To extend the approach to multiple modalities, we would require data with appropriate splits to ensure no leakage via the same time or location stamps.

3. Methodology

3.1. Data and proxy grouping

We chose the ENA24 version released officially [2] and picked four classes, containing 250 images each. All the 1,000 images that we requested were downloaded and decrypted successfully, and SHA-256 hashes of them were calculated. The total file size is 385,744,873 bytes. Images are not cropped but presented in full RGB, hence the problem is single label classification, not detection.

Since there was no data available on the sequence ID and camera ID for each image, same class images were first clustered based on their numeric IDs, where a new cluster starts whenever

there is a gap of more than three IDs or whenever the number of images in the cluster exceeds five. The downloaded images were compared using 64-bit dHash and were clustered only if the hashes matched. The total number of clusters reported is 160, which comprises of 47 bear, 49 crows, 40 dogs, and 24 opossum clusters.

3.2. Models and nested evaluation

Stage 1 uses torchvision's ImageNet-pretrained MobileNetV3-Small, and stage 2 uses ImageNet-pretrained ResNet-18 [13], [14]. Images are resized and center-cropped to 160 by 160 pixels using the corresponding weight transforms. Both backbones are frozen. Their features are standardized and classified with scikit-learn multinomial LogisticRegression using the lbfgs solver and a maximum of 1,000 iterations. The regularization value is selected from C = 0.1, 1, or 10 on model_val, after which the head is refitted on fit + model_val. Thus, the experiment measures selective use of fixed visual representations rather than end-to-end fine-tuning. For an image x, let p1(y|x) and p2(y|x) denote the calibrated class-probability vectors from stages 1 and 2. Stage 1 always runs, and stage 2 runs when the highest calibrated stage-1 class probability is below threshold t, selected on routing-validation data.

Five outer group-stratified folds are used. Each outer training partition is divided into fit, model_val, calibration, and routing_val. The model-validation split selects the logistic-regression regularization value; the calibration split fits one temperature per stage; the routing-validation split selects the cascade threshold. In the formal run, the fit partitions contain 466 to 497 images, model-validation partitions 92 to 115, calibration partitions 98 to 120, and routing-validation partitions 96 to 105. The outer test fold contains 200 images, is used once for final prediction, and never participates in model selection, calibration, or threshold selection. Thus, each image contributes exactly one outer-test prediction while the admissible selection information remains inside the corresponding training portion.

3.3. Routing and computation accounting

Within each fold, the routing rule evaluates a grid of confidence values and the observed routing-validation confidences. It keeps points within 0.02 Macro-F1 of the stage-2 score and selects the one with the lowest expected MAC cost, using deterministic tie-breaking. A threshold of zero makes no stage-2 calls. We report target budgets of 10%, 25%, 50%, and 75% only as auxiliary analyses. Their thresholds are selected on routing validation and applied unchanged to the corresponding test fold. Confidence distributions differ across folds, so nominal and observed call rates are not always identical. These auxiliary points cannot repair the primary rule after test results are observed. Stage 1 requires 28,233,152 counted Conv2d and Linear MACs, and stage 2 requires 925,286,400. One-thread CPU timing measures only frozen feature extraction with batch size 1 and 160 by 160 input. The medians are 3.686 ms for stage 1 and 25.391 ms for standalone stage 2. These measurements are computer-side cost references, not energy or complete edge-device measurements.

3.4. User-oriented workflow

The prototype uses a simple workflow. A user acquires or uploads an image and receives a calibrated MobileNet prediction. When the routing rule indicates uncertainty, the system requests ResNet verification and records the class, confidence, model, routing status, and estimated cost. Low-confidence cases can be retained for human review. Timestamps, locations, acoustic activity, and other metadata could later be added before routing, but the current implementation uses no contextual signal.

4. Experimental Setup

The formal run used seed 2026, Python 3.13.13, CPU-only PyTorch 2.12.1, torchvision 0.27.1, scikit-learn 1.9.0, NumPy 2.5.0, and Pillow 12.3.0. We use Macro-F1 as the primary metric. Balanced accuracy, NLL, 15-bin ECE, stage-2 call rate, MACs, and restricted CPU latency are secondary measures. The baselines are MobileNet-only and ResNet-only inference. Main results come from pooled out-of-fold predictions, with one outer-test prediction for each image. The table and figures are generated from the recorded JSON and CSV files. The comparison is descriptive. It does not claim statistical equivalence, significance, or generalization beyond the selected source data.

Table 1. Core pooled out-of-fold operating points. Fixed-budget results are auxiliary.

Method	Stage-2 call rate	Macro-F1	Balanced accuracy	ECE	NLL	Expected MACs	Expected latency (ms)
MobileNetV3-Small only	0.000	0.89535	0.895	0.03145	0.44299	28,233,152	3.686
Validation-selected cascade	0.039	0.89720	0.897	0.03325	0.43104	64,319,322	4.676
Auxiliary budget 50% (observed 52.2%)	0.522	0.92219	0.922	0.02819	0.26796	511,232,653	16.940
ResNet-18 only	1.000	0.92397	0.924	0.03265	0.26221	925,286,400	25.391

5. Results and Discussion

5.1. Primary cascade result

Stage 1 achieved 0.89535 Macro-F1, versus 0.92397 Macro-F1 of stage 2. The selected cascade of stage 2 selected 39 out of 1,000 images and achieved 0.89720 Macro-F1. This is 0.00185 higher than stage 1 but 0.02677 lower than stage 2. Therefore, the cascade failed to pass the predefined tolerance of 0.02. There was an operating point with a low cost; however, the future rule was not transferred robustly to the out-of-fold sets. The NLL value of the cascade was 0.43104, and the ECE value was 0.03325. The NLL and ECE values of stage 1 were 0.44299 and 0.03145, respectively, and the same values of stage 2 were 0.26221 and 0.03265.

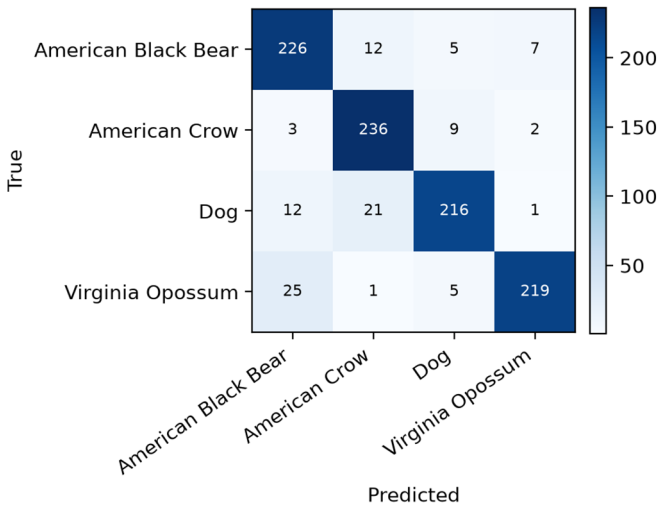

Figure 1. Pooled out-of-fold confusion matrix for the validation-selected cascade.

5.2. Auxiliary trade-off and threshold failure

The increase in fixed budget sizes made the cascade move closer to ResNet-18. For 52.2% and 74.2% call rates, we got Macro-F1 equal to 0.92219 and 0.92314 correspondingly. The points 10.0% and 22.1% gave us 0.90628 and 0.91212 correspondingly. All the points mentioned above are retrospective operating points and cannot confirm equivalence results, which would be needed to validate the unsuccessful primary selection rule. In Figure 2, you can see the curve for call rate. The latency is still available in the results file and Table 1.

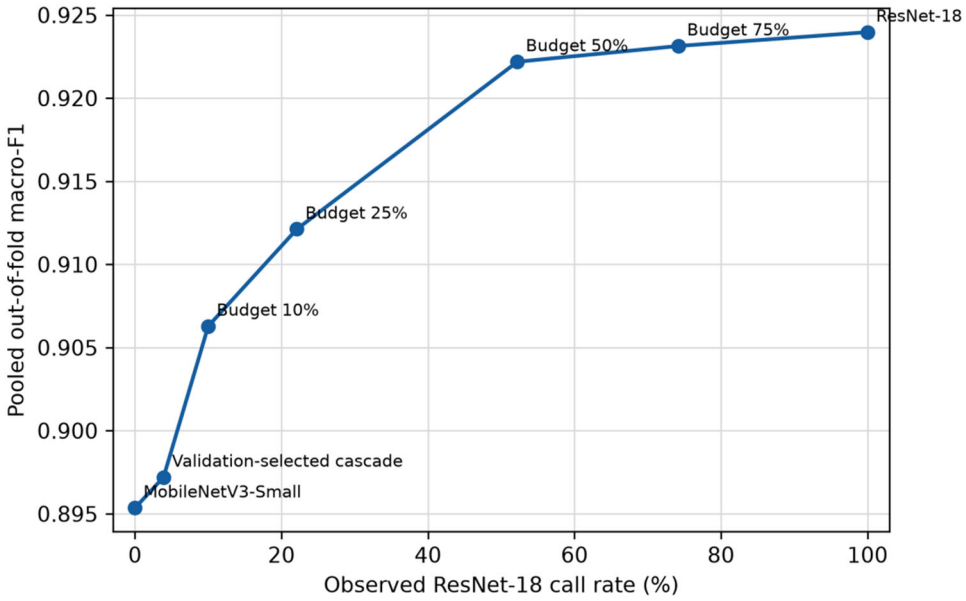

Figure 2. Auxiliary fixed-budget trade-off. The points were not selected from pooled test outcomes.

The problem probably stems from the small number of images in each routing-validation set. They had between 96 and 105 images distributed among four classes and a small number of proxies. In three of the five folds, the selection had no test calls at all. The other two had test calls of around 23% of routing validation call rate, but only 8.5% and 11% were test calls. With the low numbers of observations, there is a discrete difference in the Macro-F1 scores, and the tail of the confidence interval could move from one proxy group to another.

5.3. Implications for the prototype

These outcomes suggest an interaction at two levels, where MobileNet performs a cheap initial prediction, and ResNet verifies selectively. Since the initial rule failed to maintain stage-2 accuracy, the system prototype must reveal confidence and verification information instead of presenting all predictions as equally trustworthy. Acoustic behavior, timestamps, geographical coordinates, and user annotations could be future routing factors, particularly for multimodal systems like SmartWilds [15]. The current solution has an edge orientation instead of edge validation. The MACs and time limitation on CPU represent a fair way to compare costs, but not real measurements of device capabilities.

6. Limitations

This is a small proof of concept using 1,000 images, four classes, one source dataset, and 160 proxy groups. Proxy grouping reduces obvious burst leakage but cannot replace authoritative sequence or camera-location metadata. No cross-location or cross-dataset generalization test

was performed. Both backbones were frozen, only one seed and one five-fold split were used, and no confidence intervals or repeated nested cross-validation were computed. The threshold-selection partitions were small, so the reliability of the selected policy remains uncertain. The study did not evaluate detection, multilabel recognition, the complete ENA24 dataset, real hardware, power, energy, or end-to-end application latency. It also did not download or train on SmartWilds audio and reports no visual-audio fusion result. Conclusions apply only to the evaluated visual pipeline and should not be generalized to conditional inference or edge deployment.

7. Conclusion

We evaluated a reproducible MobileNetV3-Small/ResNet-18 cascade for resource-aware wildlife image analysis. With group-aware five-fold evaluation and separate calibration and routing-validation data, stage 1 and stage 2 reached Macro-F1 values of 0.89535 and 0.92397. The validation-selected cascade called stage 2 for 3.9% of images, reached 0.89720, and failed the 0.02 tolerance by 0.00677. The result shows that a completed cost-performance curve does not guarantee a stable prospective routing policy in a small dataset.

The resulting artifact is a bounded visual prototype for user-oriented wildlife image analysis. It records confidence and computation, while SmartWilds and Cross-Modal Corroboration show how acoustic and other sensing streams could provide future context or reliability evidence . A multimodal extension would require a separate paired-data protocol and evaluation.

8. Data and Code Availability Statement

ENA24 images and metadata are available from the official LILA ENA24 page [2]. The exact manifests, verification and grouping reports, code, configuration, result files, and Word manuscript are included in this project. REPRODUCIBILITY.md documents the environment, pretrained-weight sources, commands, outputs, and limitations. The corresponding BibTeX records are provided in paper/references.bib; no external archival identifier or public code-release URL is claimed.

References

[1] Swanson, A., Kosmala, M., Lintott, C., Simpson, R., Smith, A., & Packer, C. (2015). Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna. Scientific Data, 2, 150026. https://doi.org/10.1038/sdata.2015.26.

[2] Labeled Information Library of Alexandria: Biology and Conservation. (2019). ENA24-detection. LILA BC dataset page. https://lila.science/datasets/ena24detection

[3] Norouzzadeh, M. S., et al. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences, 115(25), E5716–E5725. https://doi.org/10.1073/pnas.1719367115

[4] Yousif, H., Yuan, J., Kays, R., & He, Z. (2017). Fast human-animal detection from highly cluttered camera-trap images using joint background modeling and deep learning classification. In 2017 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1–4). https://doi.org/10.1109/ISCAS.2017.8050762

[5] Yousif, H., Yuan, J., Kays, R., & He, Z. (2019). Animal scanner: Software for classifying humans, animals, and empty frames in camera trap images. Ecology and Evolution, 9(4), 1578–1589. https://doi.org/10.1002/ece3.4747

[6] Teerapittayanon, S., McDanel, B., & Kung, H. T. (2016). BranchyNet: Fast inference via early exiting from deep neural networks. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 2464–2469). https://doi.org/10.1109/ICPR.2016.7900006

[7] Bolukbasi, T., Wang, J., Dekel, O., & Saligrama, V. (2017). Adaptive neural networks for efficient inference. In Proceedings of the 34th International Conference on Machine Learning (Vol. 70, pp. 527–536). PMLR.

[8] Wang, X., Yu, F., Dou, Z.-Y., Darrell, T., & Gonzalez, J. E. (2018). SkipNet: Learning dynamic routing in convolutional networks. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 420–436). https://doi.org/10.1007/978-3-030-01261-8_25

[9] Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning (Vol. 70, pp. 1321–1330). PMLR.

[10] Nixon, J., Dusenberry, M. W., Zhang, L., Jerfel, G., & Tran, D. (2019). Measuring calibration in deep learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 38–41).

[11] Beery, S., Van Horn, G., & Perona, P. (2018). Recognition in terra incognita. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 472–489). https://doi.org/10.1007/978-3-030-01270-0_28

[12] Tabak, M. A., et al. (2020). Improving the accessibility and transferability of machine learning algorithms for identification of animals in camera trap images: MLWIC2. Ecology and Evolution, 10(19), 10374–10383. https://doi.org/10.1002/ece3.6692

[13] Howard, A., et al. (2019). Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (pp. 1314–1324). https://doi.org/10.1109/ICCV.2019.00140

[14] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 770–778). https://doi.org/10.1109/CVPR.2016.90

[15] Kline, J., Potlapally, A., Pillai, B., Wani, T., Patil, V., Covey, P., Katole, R., Stevens, S., Subramoni, H., Berger-Wolf, T., & Stewart, C. (2025). SmartWilds: Multimodal wildlife monitoring dataset. arXiv preprint arXiv:2509.18894 (Version 2). https://arxiv.org/abs/2509.18894v2

[16] Pillai, B., Viswapriyan, V., Stewart, C., Berger-Wolf, T., & Kline, J. (2026). Cross-modal corroboration for annotation-free wildlife monitoring. arXiv preprint arXiv:2606.21613 (Version 1). https://arxiv.org/abs/2606.21613v1

[17] Yousif, H., Kays, R., & He, Z. (2019). Dynamic programming selection of object proposals for sequence-level animal species classification in the wild. IEEE Transactions on Circuits and Systems for Video Technology.