International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 9s, 2025
https://www.theaspd.com/ijes.php

Comparative Analysis Of Various Opencv Methods For Automatic Species Detection Using Camera Trap Images

Mohammed Shameer M C[1*], Mubeena V[2]

[1*]Assistant Professor, Dept. of Computer Science, Farook College (Autonomous), Kozhikode
[2]Assistant Professor, Dept. of Vocational Studies, Farook College (Autonomous), Kozhikode

Abstract:
Biodiversity monitoring and conservation is essential for protecting ecosystem and to ensure the ecological resilience to disturbances including climatic change, disease outbreaks, and human exploitation. Camera traps is a vital tool for wildlife species monitoring that generate vast collections of images which require automated processing. Numerous approaches are available in literature. Though deep learning techniques provide higher accuracy in automated species detection, many conservation projects still employs traditional image processing techniques using OpenCV due to its lightweight nature and inherent limitation to computational resources. This paper presents the first comprehensive, head-to-head evaluation of six OpenCV based species detection techniques including three classical methods (Background Subtraction - Contour Analysis, Haar-Cascade, HOG + SVM) and three deep learning based methods (MobileNet-SSD, YOLOv5-ONNX, EfficientDet-D0) across three publicly available camera trap datasets (Snapshot Serengeti, Caltech Camera Traps and CamTrapAsia). Results show that YOLOv5-ONNX achieves the highest mean Average Precision (mAP = 93.4%). The classical Background Contour method still remains effective for large species (elephants) (F1 = 0.79) while running faster on Raspberry Pi 4 hardware. This study highlights a trade-offs in accuracy, inference speed, energy footprint, and data requirements, providing actionable guidelines for biologists in selecting OpenCV pipelines under real - world conditions.

Keywords: *Camera traps; Species detection; OpenCV; YOLOv5; Haar cascade; Wildlife monitoring; Performance benchmarking.*

1. Introduction

Biodiversity conservation is very important to ensure the stability and sustainability of the earth's ecological balance. Each species plays an important role in the ecosystem. As ecosystems face exceptional stress due to human interventions like deforestation, urbanization, loss of habitat and climate change, it is essential to track how species or specific animal populations respond and adapt to these changes [1]. Such monitoring helps early detection of species population decline, guides habitat protection and conservation efforts, and reforms environmental policies. Thus biodiversity monitoring is essential for protecting ecosystem and to ensure the ecological resilience [2].

Animal resource investigation is an important technique used in biodiversity conservation that provides insight into the species presence, their movement and behavior. Camera trapping is the most widely used technique in evidence based conservation [3][4]. This technique can be used to identify biodiversity hotspots and endanger species that require immediate actions. Camera traps generates huge volume of data and require much effort to sort, analyze and annotate the data [5].

Many automated image processing techniques are available for handling the camera trap images such as deep neural networks (DNNs). DNNs yields 90% accuracy for large and balanced data sets. As many of the field deployments are operated utilizing battery-operated edge devices where computational resources, connectivity, and annotated training data are deficient [6]. OpenCV provide extensive methods of traditional and deep neural network (DNN) modules that are capable of being executed on economically viable hardware; however, there exists a dearth of empirical evidence regarding which OpenCV methodology most effectively reconciles performance with resource utilization in the context of automatic species detection.

This study compare and analyze the performance of three classical (Background Subtraction- Contour Analysis, Haar-Cascade, HOG- SVM) and three deep learning (MobileNet-SSD, YOLOv5-ONNX, EfficientDet-D0)

image processing techniques across three public camera trap datasets (Snapshot Serengeti , Caltech Camera Traps and CamTrapAsia). Major contributions of this study include:
1. Designing a reproducible evaluation protocol for comparing six OpenCV techniques on three camera trap datasets.
2. Produce a comprehensive metrics (mAP, precision, recall, F1) and analyze per species and day night variation.
3. Provide practical recommendations and decision matrices to help practitioners match OpenCV pipelines to deployment scenarios.

Remaining sections of this paper is organized as follows. Section 2 discuses the species detection methods available in literature. Section 3 describes the details of the data set used for this study. Section 4 highlights the methodology and section 5 discuses the experimental set up. A detailed analysis and discussion of the result is given in section 6 and section 7 concludes the study.

2. Literature Review

There are mainly two broad categories of approaches for automated species detection from camera trap images - Classical computer vision methods and deep learning based techniques. Both of these techniques aim to enhance biodiversity monitoring by automating the identification and classification of species using large volume of captured images, thus addressing the challenges posed by large datasets.

2.1 Traditional Methods

Early wildlife - monitoring systems relied on background subtraction for detecting moving objects in video sequences. Here frames are compared with static background model to identify changes that indicate movements [7][8]. To cope with the environmental variations and light conditions, the model should be enhanced with additional color and gradients features [9][10]. The Histogram of Oriented Gradients (HOG) along with Support Vector Machine (SVM) is another approach used for object detection. HOG is capable of capturing edge and gradient details and SVM classifies these features to identify the species [11][12]. This method is sensitive to background and light conditions and has achieved moderate success. Haar cascade classifier method is used for rapid object detection in real-time applications. This approach utilizes a series of simple features to identify each object. They are suitable for wildlife detection in dynamic environments [13].

2.2 Deep Learning based Approaches

Recent studies demonstrate the use of deep learning models to localize and classify species. Such methods and methods using convolutional neural networks (CNNs) achieve higher performance in species recognition [6]. YOLOv5m has achieved a promising accuracy of 97 to 99 % on night-time datasets [14]. YOLOv10-X enables counting of individual images [15][16]. Fine tuning and fusing temporal metadata can further enhance accuracy [15]. The DNN module of The OpenCV package supports inference for ONNX and TensorFlow models, thus enabling edge deployment of compact CNNs such as MobileNet-SSD.

Comprehensive Models that make use of CameraTrapDetectoR package offers models that can classify various species, achieving a mean average precision values between 0.80 and 0.96. This facilitates more user-friendly applications for conservators and biologists [17][18].

Combining deep learning techniques with contextual data such as habitat information, vegetation type and time of day, provide enhanced ecological insights. Such advanced systems supporting more accurate wildlife management decisions [16][19].

Use of automated methods significantly reduce the need of manual effort and intervention required for species identification, still there exist concern regarding which method is more suitable for a specific ecosystem. Balanced automations with expert insights will be crucial for effective conservation strategies and decisions [20]. This decision requires ecological understanding. Further, only few studies attempted compare and analyze the classical and deep learning techniques under uniform conditions. This study work addresses this gap with a holistic, resource aware benchmark.

3. Datasets
This study leverages on the three widely used and open source datasets - Snapshot Serengeti (SS) [21], Caltech Camera Traps (CCT-20) [22] and CamTrapAsia [23]. While SS utilizes public participation to classify images, CCT 20 makes use of advanced algorithms for species identification and analysis from camera trap data with a different methodological approach. CamtrapAsia focuses on biodiversity monitoring in Asian ecosystems using camera trap technology to gather data on wildlife utilizing local participation. All datasets include bounding box annotations. Features of the three dataset are tabulated as in Table 1.

Dataset	Geography	Images	Species	Day/Night(%)	Annotation Format
Snapshot Serengeti (SS)	Tanzania	7.1 million	61	62/38	JSON
Caltech Camera Traps (CCT-20)	USA	243,100	20	75 / 25	JSON
CamTrapAsia	Tropical Asia, including Indonesia, Malaysia	278,260	371	55 / 45	JSON/XML

Table 1: Details of the dataset used

4. Methodology
4.1 Traditional Methods
(a) The Background Contour (BG-CT) methodology is based on Mixture of Gaussians (MOG2) algorithm for background subtraction that can isolate forward moving objects in a static environment. This method is very efficient for images captured in daylight where background is relatively stable. Morphological opening is performed to remove minor artifacts and noises through erosion and dilation. The resultant binary mask is then processed to extract the contours of the objects. Each contour is later analyzed according to area and aspect ratio to exclude non-animal objects. (b) The Haar Cascade classifier is a feature-based detection approach that utilizes trained cascade classifiers to identify objects. Here the system is trained with 6,000 positive and 9,000 negative samples with images resized to dimensions of 144 × 144 pixels. For feature extraction, Local Binary Patterns (LBP) algorithm is used. (c) The HOG + SVM approach utilizes Histogram of Oriented Gradients (HOG) descriptor that effectively captures local edge orientation and intensity within a detection window of 64 × 128 pixels. The captured image is partitioned into cells of 8×8 pixel dimension, and 9 bin histogram is calculated for each cell. Block normalization is performed to augment light variance. The resulting descriptor is used for training the linear Support Vector Machine (SVM) classifier. The model underwent iterative refinement through three rounds of hard negative mining.

4.2 Deep Learning based Approaches
(a) The MobileNet SSD model is a lightweight, real-time object detector trained in TensorFlow Lite using int8 quantization for efficient inference on edge devices. Here, it is converted to OpenCV's DNN module to enable direct deployment across platforms. The model features a streamlined architecture that balances detection accuracy and speed, making it suitable for embedded applications. (b) The YOLOv5 ONNX utilizes the small variant of Ultralytics' YOLOv5 model, pre-trained on the dataset and subsequently fine-tuned for 50 epochs on each of the three datasets (SS, CCT-20, and CamTrapAsia). Once exported to ONNX format, this model enables hardware-agnostic acceleration through OpenCV's backend. (c) The EfficientDet D0 (E-D0) model is a reimplementation of EfficientDet Lite0 with an input resolution of 512 × 512 pixels. This model designed for mobile and edge scenarios. It uses compound scaling and efficient BiFPN feature fusion, offering strong performance at low computational cost.
All three models were integrated and evaluated uniformly within OpenCV's inference framework to ensure fair comparisons.

4.3 Training
The deep learning models used in this study is trained using the Stochastic Gradient Descent (SGD) optimization algorithm with a learning rate (lr = 0.001) and a momentum value of 0.9. These values are chosen

to provide a stable convergence during the training process. A consistent batch size of 32 is used in the experiments to balance between memory usage and better gradient stability. Training and fine-tuning were performed on an NVIDIA RTX A6000 GPU system. The classical computer vision approaches discussed in this paper does not require any GPU acceleration and were executed entirely on the CPU due to their lightweight nature. To achieve optimal performance from each model, a 5-fold random search strategy is carried out for hyper-parameter tuning. This involve randomly sampling hyper-parameter combinations within defined ranges over five validation folds, allowing for healthy generalization and minimization of over-fitting across the datasets. The hyper-parameter search grid is given in Table 2.

Model	Dataset	Learning Rate	Batch Size	Momentum	Window Size / Anchor Scale	Best Value Achieved
BG - CT	SS	N/A	N/A	N/A	Area > 1000 px	F1 = 0.74
	CCT-20	N/A	N/A	N/A	Area > 500 px	F1 = 0.69
	CamTrapAsia	N/A	N/A	N/A	Area > 1000 px	F1 = 0.79
Haar - Cascade	SS	NA	NA	NA	144×144 patch, neighbors=4	F1 = 0.67
	CCT-20	NA	NA	NA	144×144 patch, neighbors=4	F1 = 0.64
	CamTrapAsia	NA	NA	NA	144×144 patch, neighbors=4	F1 = 0.65
HOG + SVM	SS	NA	NA	NA	64×128, stride=8	F1 = 0.61
	CCT-20	NA	NA	NA	64×128, stride=8	F1 = 0.59
	CamTrapAsia	NA	NA	NA	64×128, stride=8	F1 = 0.63
MobileNet - SSD	SS	0.001–0.01	32	0.9	0.35 - 0.75	mAP = 85.2%
	CCT-20	0.001–0.01	32	0.9	0.35 - 0.75	mAP = 86.5%
	CamTrapAsia	0.001–0.01	32	0.9	0.35 - 0.75	mAP = 85.6%
YOLOv5 - ONNX	SS	0.0005 -0.005	32	0.9	Anchor = auto	mAP = 93.4%
	CCT-20	0.0005 - 0.005	32	0.9	Anchor = auto	mAP = 91.2%
	CamTrapAsia	0.0005–0.005	16–64	0.9	Anchor = auto	mAP = 94.6%
EfficientDet - D0	SS	0.0005 - 0.005	16	0.9	Input : 384 - 512px	mAP = 92.3%
	CCT-20	0.0005 -0.005	16	0.9	Input : 384 - 512px	mAP = 89.9%
	CamTrapAsia	0.0005–0.005	16–64	0.9	Input : 384 - 512px	mAP = 90.7%

Table 2: Hyper-parameter search grid

5. Experimental Setup

To evaluate the performance of species detection model, following setup is carried out. The training is carried out in high-performance workstation with an Intel Core i7-13900K processor, 64 GB of RAM, and an NVIDIA RTX A6000 GPU. This configuration ensures fast model training and efficient handling of large datasets with high-resolution input images. For carrying out edge inference test, the Raspberry Pi 4 with 4 GB of RAM and the NVIDIA Jetson Nano devices are used. These devices are resource constrained, low-power platforms suited for real-world, low-resource environments.

The metrics used to evaluate the Model performance include mean Average Precision at a threshold of 0.5 (mAP = 0.5), precision, recall, and F1-score. This metrics represents a balanced view of detection accuracy and robustness of the system.

70% percent of the images in the dataset is used for training purpose and 15% for validation and remaining for testing (15%). Each experiment was repeated three times per model, and the final results are reported as the mean values ensuring statistical reliability of the findings.

6. Results and Discussions

The result of the experiment carried out shows deep learning models outperforms the classical model in accuracy and precision. The species specific F1 and Recall values are tabulated for each dataset is given in Table 3, Table 4 and Table 5.

Species	BG-CT		Haar Cascade		HOG + SVM		MobileNet - SSD		YOLOv5 - ONNX		EfficientDet - D0	
	F1	Recall	F1	Recall	F1	Recall	F1	Recall	F1	Recall	F1	Recall
Zebra	0.72	0.65	0.67	0.62	0.61	0.58	0.85	0.84	0.96	0.94	0.94	0.90
Wildebeest	0.74	0.65	0.65	0.61	0.60	0.56	0.86	0.85	0.95	0.92	0.93	0.90
Elephant	0.69	0.64	0.63	0.59	0.59	0.54	0.82	0.81	0.93	0.91	0.91	0.89

Table 3: Per-Species F1 and Recall Scores for Snapshot Serengeti Dataset

Species	BG-CT		Haar Cascade		HOG + SVM		MobileNet - SSD		YOLOv5 - ONNX		EfficientDet - D0	
	F1	Recall	F1	Recall	F1	Recall	F1	Recall	F1	Recall	F1	Recall
Mule Deer	0.69	0.64	0.64	0.61	0.59	0.56	0.83	0.80	0.91	0.91	0.89	0.87
Coyote	0.65	0.61	0.61	0.56	0.57	0.54	0.81	0.80	0.90	0.88	0.88	0.86
Bobcat	0.62	0.58	0.64	0.59	0.55	0.51	0.79	0.78	0.89	0.88	0.87	0.86

Table 4: Per-Species F1 and Recall Scores for Caltech Camera Traps Dataset

Species	BG-CT		Haar Cascade		HOG + SVM		MobileNet - SSD		YOLOv5 - ONNX		EfficientDet - D0	
	F1	Recall	F1	Recall	F1	Recall	F1	Recall	F1	Recall	F1	Recall
Elephant	0.79	0.73	0.65	0.60	0.63	0.57	0.86	0.82	0.96	0.95	0.95	0.92
Leopard	0.66	0.69	0.61	0.56	0.56	0.56	0.88	0.84	0.95	0.92	0.94	0.90
Civet	0.34	0.29	0.42	0.36	0.39	0.36	0.78	0.75	0.92	0.89	0.88	0.85

Table 5: Per-Species F1 and Recall Scores for CamTrapAsia Dataset

The deep learning model YOLOv5 ONNX outperforms all other models across the three dataset. YOLOv5 achieves the highest mean Average Precision of 93.4% on SS dataset, 91.2% on CCT-20 dataset and 94.6% on the CamTrapAsia dataset respectively a with threshold value of 0.5. These results highlight robustness and adaptability of YOLOv5 deep learning model after fine-tuning for 50 epochs on each dataset. YOLOv5 showed higher performance in day-night parity as well, aligning with earlier findings. The day-night performance of the YOLOv5 is tabulated as in Table 6.

Dataset	Day/Night	Precision	Recall	mAP
SS	Day	0.936	0.924	0.931
	Night	0.906	0.893	0.902
CCT-20	Day	0.928	0.917	0.918
	Night	0.902	0.884	0.908
CamTrapAsia	Day	0.945	0.935	0.946
	Night	0.926	0.912	0.937

Table 6: Day - Night Accuracy for YOLOv5-ONNX on all datasets

EfficientDet - D0 model is another reliable species detection model with mean Average Precision of 92.3%, 89.9% and 90.7% on SS, CCT-20 and CamTrapAsia datasets respectively.

Traditional approaches BG-CT based on background subtraction marked the lowest performance in terms of mAP among all models. But this model achieved a F1 Score of 0.79 (elephant) shows its effectiveness in detecting large species. Though BG CT model struggle in detecting smaller and overlapping species, it can be effectively utilized for detecting lager species with minimal cost. Another notable finding from the study is that, deep learning model MobileNet-SSD showed its weakness in identifying smaller species but excelled for medium and larger species identification.

7. Conclusion

This empirical study is an attempt to evaluate the performance and behavior of OpenCV methods for automatic species detection using camera trap images. OpenCV package provide two categories of automated

species detection – using classical and deep learning based approaches. Study underlines the performance advantages of deep learning methods in automated species detection. YOLOv5 method marked the highest precision value across all dataset and is most suited for identifying small and large species. The classical approaches are still relevant in identifying large species and can be utilized for resource constraint environments where cost matters.
This study is based on the publically available dataset that covers limited geographical areas. As OpenCV package does not support fusing temporal data, future studies should consider incorporating those with self supervised pre- training and optimizations to improve the accuracy.

References

[1] Craig P. Cordier, David A. Ehlers Smith, Yvette Ehlers Smith, Colleen T. Downs, Camera trap research in Africa: A systematic review to show trends in wildlife monitoring and its value as a research tool, Global Ecology and Conservation, Volume 40, 2022, https://doi.org/10.1016/j.gecco.2022.e02326.
[2] Z.J. Delisle, M. Henrich, P. Palencia, R.K. Swihart, Reducing bias in density estimates for unmarked populations that exhibit reactive behaviour towards camera traps, Methods Ecol. Evol., 14 (2023), pp. 3100-3111, https://doi.org/10.1111/2041-210X.14247
[3] S. Gutierrez-Zapata, S. Santoro, M.E. Gegundez-Arias, N. Selva, J. Calzada, Dog invasions in protected areas: a case study using camera trapping, citizen science and artificial intelligence, Global Ecol. Conserv., 54 (2024), Article e03109, https://doi.org/10.1016/j.gecco.2024.e03109
[4] D. Velasco-Montero, J. Fernández-Berni, R. Carmona-Galán, A. Sanglas, F. Palomares, Reliable and efficient integration of AI into camera traps for smart wildlife monitoring based on continual learning, Eco. Inform., 83 (2024), Article 102815, 10.1016/j.ecoinf.2024.102815
[5] D.L. Norman, P.H. Bischoff, O.R. Wearn, R.M. Ewers, J.M. Rowcliffe, B. Evans, S. Sethi, P.M. Chapman, R. Freeman, Can CNN-based species classification generalise across variation in habitat within a camera trap survey? Methods Ecol. Evol., 14 (2023), pp. 242-251, 10.1111/2041-210X.14031
[6] D.-Q. Yang, G.-P. Ren, K. Tan, Z.-P. Huang, D.-P. Li, X.-W. Li, J.-M. Wang, B.-H. Chen, W. Xiao, An adaptive automatic approach to filtering empty images from camera traps using a deep learning model Wildl. Soc. Bull., 45 (2021), pp. 230-236, 10.1002/wsb.1176
[7] Raji, C. G. (2023). Early Warning System from Threat of Wild Animals using Digital Image Processing. 322–326. https://doi.org/10.1109/ICSMDI57622.2023.00064
[8] Han, B., & Davis, L. S. (2012). Density-Based Multifeature Background Subtraction with Support Vector Machine. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(5), 1017–1023. https://doi.org/10.1109/TPAMI.2011.243
[9] Xu, Y., Xu, L., Li, D., & Wu, Y. (2011). Pedestrian detection using background subtraction assisted Support Vector Machine. Intelligent Systems Design and Applications, 837–842. https://doi.org/10.1109/ISDA.2011.6121761
[10] Ahmed, A. H., Kpalma, K., & Guédi, A. O. (2017). Human Detection Using HOG-SVM, Mixture of Gaussian and Background Contours Subtraction. Signal-Image Technology and Internet-Based Systems, 334–338. https://doi.org/10.1109/SITIS.2017.62
[11] Han, G., Gu, J., Li, X., & Wu, P. (2015). Moving object detecting method based on background subtraction and HOG features.
[12] Zhou, D., Wang, J., & Wang, S. (2012). Countour based HOG deer detection in thermal images for traffic safety. International Conference on Image Processing, 969–974.
[13] Ashwini, B., Yuvaraju, B. N., Pai, A. Y., & Baliga, B. A. (2017). Real Time Detection and Classification of Vehicles and Pedestrians Using Haar Cascade Classifier with Background Subtraction. https://doi.org/10.1109/CSITSS.2017.8447818
[14] Yu, X., Wang, J., Kays, R., Jansen, P. A., Wang, T., & Huang, T. S. (2013). Automated identification of animal species in camera trap images. Eurasip Journal on Image and Video Processing, 2013(1), 52. https://doi.org/10.1186/1687-5281-2013-52

[15] Loos, A., Weigel, C., & Koehler, M. (2018). Towards Automatic Detection of Animals in Camera-Trap Images. European Signal Processing Conference, 1805–1809. https://doi.org/10.23919/EUSIPCO.2018.8553439

[16] Fergus, P., Chalmers, C., Matthews, N., Nixon, S., Burger, A., Hartley, O., Sutherland, C., Lambin, X., Longmore, S. N., & Wich, S. A. (2024). Towards Context-Rich Automated Biodiversity Assessments: Deriving AI-Powered Insights from Camera Trap Data. https://doi.org/10.48550/arxiv.2411.14219

[17] Tabak, M. A., Falbel, D., Hamzeh, T., Brook, R. K., Goolsby, J. A., Zoromski, L. D., Boughton, R. K., Snow, N. P., VerCauteren, K. C., & Miller, R. S. (2022). CameraTrapDetectoR: Automatically detect, classify, and count animals in camera trap images using artificial intelligence. bioRxiv. https://doi.org/10.1101/2022.02.07.479461

[18] Giraldo-Zuluaga, J.-H., Salazar, A., Gomez, A., & Diaz-Pulido, A. (2017). Automatic Recognition of Mammal Genera on Camera-Trap Images using Multi-Layer Robust Principal Component Analysis and Mixture Neural Networks. arXiv: Computer Vision and Pattern Recognition. https://arxiv.org/abs/1705.02727

[19] Thom, H. (2017). Unified detection system for automatic, real-time, accurate animal detection in camera trap images from the arctic tundra. https://munin.uit.no/handle/10037/11218

[20] Gomez, A., Salazar, A., & Vargas, F. (2016). Towards Automatic Wild Animal Monitoring: Identification of Animal Species in Camera-trap Images using Very Deep Convolutional Neural Networks. arXiv: Computer Vision and Pattern Recognition. http://export.arxiv.org/pdf/1603.06169

[21] Swanson AB, Kosmala M, Lintott CJ, Simpson RJ, 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.

[22] Caltech Camera Traps. Caltech camera traps. Retrieved from http://lila.science/datasets/caltech-camera-traps

[23] Mendes, Calebe P.,Wido R. Albert, Zachary Amir, Marc Ancrenaz,Eric Ash, Badrul Azhar, Henry Bernard, et al.2024. "CamTrapAsia: A Dataset of Tropical ForestVertebrate Communities from 239 CameraTrapping Studies." Ecology 105(6): e4299. https://doi.org/10.1002/ecy.4299ECOLOGY 3 of 3