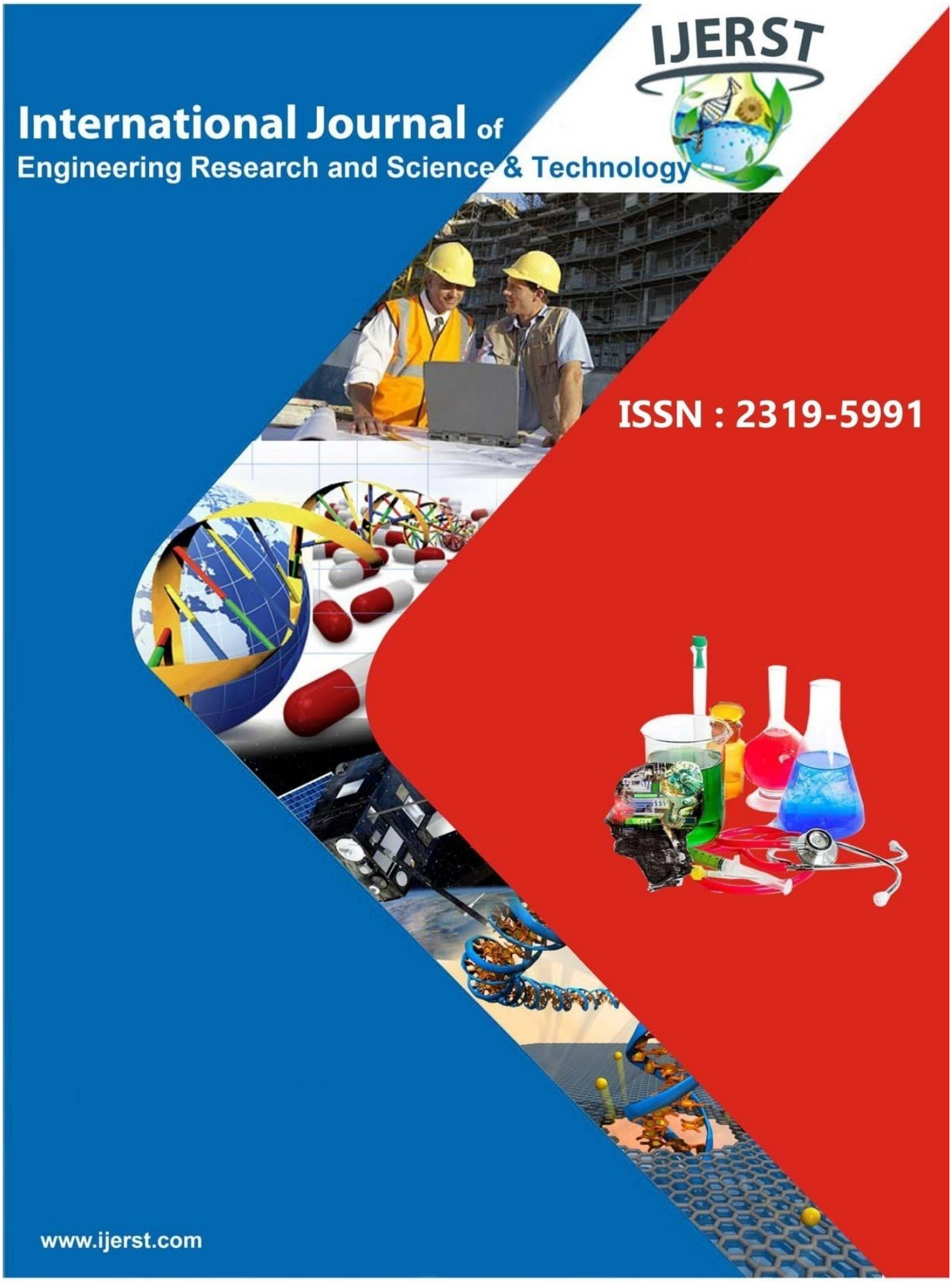

Email: editor@ijerst.com or editor.ijerst@gmail.com

ISSN 2319-5991 www.ijerst.com
Vol. 21, Issuse.1Jan 2025

BIO-EYE: AN ADVANCED WILDLIFE DETECTION FRAMEWORK

[1]K.Ramya Laxmi,[2]Maganti Nikhil Reddy,[3]Patnam Akshitha Reddy,[4]Pagadala Sandeepa,[5]Lingamgunta Hemanth

[1]Assistant Professor,Department Of Computer Science And Engineering,Sreyas Institute Of Engineering And Technology, ramya.kunta@sreyas.ac.in

[2,3,4,5]Students,Department Of Computer Science And Engineering,Sreyas Institute Of Engineering And Technology.

[2]magantinikhilreddy16@gmail.com, [3]akshithapatnam444@gmail.com,
[4]sandeepapagadala1583@gmail.com, [5]lingamguntahemanth@gmail.com.

ABSTRACT

Efficient wildlife monitoring is essential for conservation and management, with camera traps being a key tool for unobtrusive and continuous data collection. However, the manual processing of vast amounts of camera trap imagery is labor-intensive and time consuming. This paper proposes a framework utilizing deep convolutional neural networks (CNNs) for automated species recognition in the wild, based on a dataset from the Wildlife Spotter project. The system uses advanced graph-cut algorithms to segment images and identifies species automatically, significantly outperforming traditional methods like the bag of visual words model. This approach not only accelerates research and enhances citizen science initiatives but also improves management decisions. While there are challenges related to environmental conditions and species diversity, the proposed method marks a substantial advancement in automated wildlife monitoring. Integrating deep learning with camera trap technology promises to revolutionize ecological research, offering more accurate and efficient species identification and facilitating timely conservation actions.

KEYWORDS: *Camera Traps, CNN, Wildlife Spotter, Graph-Cut algorithm, Biodiversity, Species Detection, Sustainable conservation*.

I.INTRODUCTION

Wildlife monitoring plays a critical role in biodiversity conservation, ecosystem management, and ecological research. Effective monitoring provides essential data for understanding species distribution, population trends, and habitat use. Camera traps have become a vital tool in wildlife research, allowing for the unobtrusive and continuous collection of imagery from remote and often inaccessible areas. These devices capture thousands of images, providing a wealth of information that can aid conservationists and researchers. However, the manual processing and analysis of the vast amounts of data generated by camera traps are labor-intensive, time-consuming, and prone to human error. This bottleneck has led to the exploration of automated methods to streamline the identification process. The proposed wildlife detection framework addresses these challenges by leveraging the power of deep convolutional neural networks (CNNs) for automated species recognition. This framework, based on data from the Wildlife Spotter project, employs advanced graph-cut algorithms for image segmentation and species identification, significantly enhancing the efficiency and accuracy of wildlife monitoring compared to traditional methods like the bag of visual words model. By integrating deep learning techniques with camera trap technology, this framework not only accelerates research but also empowers citizen science initiatives and supports better management decisions. Despite challenges posed by varying environmental conditions and the diversity of species, the system represents a significant advancement in automated wildlife monitoring. It promises to revolutionize ecological research by providing more accurate, efficient species identification and facilitating timely conservation actions, ultimately contributing to more effective wildlife conservation and management strategies.

II.EXISTING SYSTEM

The traditional approach to species recognition in wildlife monitoring relies heavily on manual processing and basic machine learning methods. Camera traps collect large volumes of images and videos continuously, capturing animal activity unobtrusively in their natural habitats. However, manually sorting and classifying these images is highly labor-intensive, time-consuming, and prone to human error. To automate some of this process, simpler machine learning techniques, like the Bag of Visual Words (BoVW) model, have been used. This model extracts features using methods like SIFT or SURF to classify images. Although the BoVW model aids in classification, it suffers from limited accuracy and struggles to scale effectively with large datasets. As a result, the existing system is inefficient and lacks the technological sophistication needed to accurately and efficiently handle vast amounts of wildlife imagery. Here are some of the most prominent ones:

1. Manual Image Classification

Manual image classification relies on human experts to review and label camera trap images, ensuring high accuracy when done by skilled personnel. However, this method is labor-intensive, time-consuming, and prone to human error, especially as the volume of images increases. It becomes impractical for large-scale projects and is difficult to maintain consistently over time, making it inefficient for long-term wildlife monitoring.

2. Bag of Visual Words (BoVW)

The **Bag of Visual Words (BoVW)** method extracts features like **SIFT** or **SURF** from images, representing them as visual words for classification. While automating classification, BoVW struggles with environmental variations like lighting and animal pose. Furthermore, the method struggles with large datasets, as the feature extraction and clustering process can be computationally expensive and slow. Its performance often diminishes with increasing environmental variation—such as changes in lighting, background, or animal pose—which is common in wildlife monitoring. Thus, while BoVW provides a foundation for automated classification, it lacks the flexibility and accuracy required for large-scale, real-world applications.

3. Template Matching

Template matching compares captured images to predefined templates of species. It works well when conditions like background and pose are consistent. However, wildlife images often feature significant variations, making template matching unreliable for dynamic environments. It struggles with different orientations, lighting, and backgrounds, limiting its effectiveness in wildlife monitoring. As a result, template matching fails when animals are captured in different orientations, with changing backgrounds or under varying lighting conditions. While simple to implement, template matching becomes inefficient and inaccurate in dynamic environments like wildlife habitats.

4. Support Vector Machines (SVM)

SVM is a supervised machine learning algorithm that classifies images based on feature separation in a high-dimensional space. While effective for smaller datasets, SVM struggles with scalability as datasets grow, requiring substantial computational resources. It also can suffer from inaccuracies when dealing with noisy data or complex environmental conditions, limiting its efficiency in large-scale wildlife monitoring projects. Additionally, SVMs are sensitive to noisy or irrelevant features, which can lower their accuracy, especially when images come from diverse environments with varying backgrounds and lighting. While effective for smaller datasets, SVM struggles to provide the speed, scalability, and robustness required for large, dynamic wildlife datasets.

5. K-Nearest Neighbors (K-NN)

K-NN classifies images by comparing features to those of nearby images in a training set. While simple, K-NN is computationally expensive and inefficient for large datasets, as it requires comparing each test image to all training images. It also struggles with irrelevant features, reducing accuracy, and is not scalable for large wildlife monitoring projects where quick, accurate classification is essential. Furthermore, K-NN is highly sensitive to irrelevant or noisy features, which can lead to inaccurate classification if the feature extraction process is not carefully optimized. As datasets increase in size and complexity, K-NN struggles to scale effectively, making it unsuitable for large-scale wildlife monitoring tasks where rapid, accurate classification is essential.

III.PROPOSED SYSTEM

Objective of the proposed system is to overcome the major limitations of existing wildlife monitoring methods, such as inefficiency, inaccuracy, and scalability issues. By utilizing Deep Convolutional Neural Networks (CNNs), the system aims to automate species recognition, offering more accurate and scalable classification of camera trap images. It incorporates graph-cut algorithms for precise image segmentation, reducing human error and enhancing reliability. The system reduces human error, supports citizen science, and accelerates decision-making with real-time data processing. Its ability to handle large datasets and provide continuous, reliable monitoring makes it a powerful tool for enhancing wildlife research and conservation efforts. Ultimately, the proposed system significantly improves the speed, accuracy, and effectiveness of wildlife monitoring.

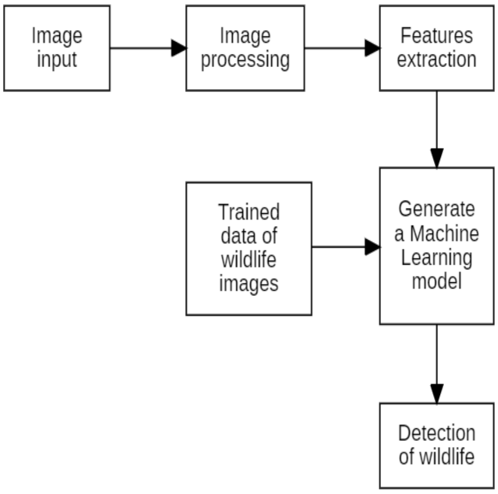

Fig 1: Architecture Diagram

The BIO-EYE project methodology adopts a structured, systematic approach to address wildlife species identification challenges through deep learning and image segmentation. It involves several phases, starting with **Requirement Gathering and Analysis**. This phase identifies system needs through stakeholder interviews with wildlife researchers, conservationists, and domain experts, defining both functional (automated species detection, real-time processing) and non-functional requirements (high accuracy, scalability). The design also includes data, hardware, and software specifications, such as the use of Python, TensorFlow, and OpenCV for model training and segmentation tasks. Additionally, it focuses on understanding the image dataset needs and ensures that the hardware setup supports efficient model training (such as GPUs and storage).

The next phase, **System Design**, focuses on modular architecture, including image preprocessing, deep learning models (such as CNN), and segmentation using graph-cut algorithms. It also covers the design of the user interface (UI) and database if applicable. A key aspect of this phase is selecting the appropriate architecture for the deep learning model, such as ResNet or Inception, and defining the process for classifying wildlife species from camera trap images. In terms of segmentation, graph-cut algorithms are integrated to efficiently separate wildlife from the background while ensuring real-time image processing capabilities. This phase also includes designing the user interface, making it intuitive and accessible for researchers, allowing them to upload images and view results seamlessly. If a database is necessary, this phase will also address the creation of a schema to store metadata related to species recognition.

During the **Development** phase, the system is implemented based on the design specifications. Image preprocessing scripts are developed to resize, filter, and augment the dataset, applying techniques like noise reduction and rotation to enhance model performance. The core CNN model is built using frameworks like TensorFlow and Keras, initially using a pre-trained model (e.g., ResNet) and fine-tuning it for the wildlife detection task. The model is then trained and evaluated based on performance metrics like accuracy, precision, recall, and F1-score. The segmentation algorithms are implemented using graph-cut techniques, ensuring accurate wildlife-background separation. If a web interface is developed, a simple UI is built using Flask or Django, integrating with the backend to allow real-time image uploading and result viewing.

In the **Testing** phase, the system undergoes rigorous testing to ensure all components function as expected. Unit testing is performed on individual modules like preprocessing, CNN model, and segmentation algorithms. Integration testing follows to assess how well the modules interact, ensuring smooth data flow and synchronization between image preprocessing, model prediction, and segmentation tasks. Performance testing evaluates system speed and efficiency, ensuring the system can handle large datasets and process images in real-time. Accuracy testing is crucial to assess the model's performance on unseen test datasets, validating results with real wildlife images. Finally, **User Acceptance Testing (UAT)** is conducted to gather feedback from wildlife researchers and conservationists, ensuring the system is user-friendly and meets their functional needs.

Deployment involves launching the system in a real-world setting, where it can begin processing wildlife camera trap images and delivering species recognition results. The model is deployed on cloud platforms (such as AWS or Google Cloud), enabling real-time image processing. If a web interface is part of the system, it is hosted on a server (e.g., Heroku, AWS), ensuring users can interact with it without issues. The backend system is integrated with APIs to handle requests between the frontend and backend, enabling seamless real-time species classification and segmentation. Monitoring tools are also implemented to track system performance, ensuring it operates efficiently and meets scalability requirements.

After deployment, the **Maintenance and Support** phase ensures the system remains functional and continuously improves. Regular model retraining with new wildlife camera trap images helps improve accuracy and adapt to newly discovered species. Bug fixes are performed based on user feedback and system monitoring. The software libraries and dependencies are updated to ensure compatibility with evolving platforms and security protocols. User support is provided, offering assistance for any technical issues encountered by users. The system is periodically optimized to handle large datasets more efficiently, and the documentation is kept up-to-date, including deployment guidelines and user manuals.

The methodology emphasizes several key considerations: data quality, model selection, segmentation accuracy, performance optimization, and scalability. High-quality annotated datasets are prioritized, and data augmentation is used to improve model robustness. The selection of pre-trained models, such as ResNet, ensures efficient species classification. Graph-cut algorithms for segmentation are optimized for both accuracy and computational efficiency, crucial for real-time processing. Scalability is achieved by utilizing cloud infrastructure, ensuring the system can handle growing datasets and increasing user demands. Ethical concerns, such as secure handling of wildlife data and responsible AI usage, are also carefully considered.

The implementation of the BIO-EYE framework integrates advanced image preprocessing techniques, a robust CNN architecture, and effective training strategies. By leveraging cutting-edge tools like TensorFlow, Keras, and OpenCV, the system achieves high accuracy in wildlife species recognition. It not only contributes to ecological monitoring but also supports conservation efforts, enabling researchers to identify species in remote, hard-to-reach areas. Furthermore, by combining these technologies with real-time capabilities and ease of use, the BIO-EYE system paves the way for more efficient, automated wildlife monitoring solutions that can be expanded for global conservation needs.

ID	TC-1	TC-2
Text Description	Test with an image of a Cow	Test with an image of a Tiger
Output		

Fig2: output

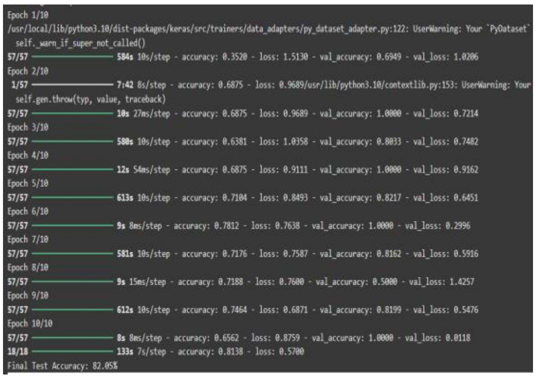
```
Epoch 1/10
/usr/local/lib/python3.10/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:122: UserWarning: Your `PyDataset`
  self._warn_if_super_not_called()
57/57 ━━━━━━━━━━━━━━━━━━━━ 584s 10s/step - accuracy: 0.3520 - loss: 1.5130 - val_accuracy: 0.6949 - val_loss: 1.0206
Epoch 2/10
 1/57 ━━━━━━━━━━━━━━━━━━━━ 7:42 8s/step - accuracy: 0.6875 - loss: 0.9689/usr/lib/python3.10/contextlib.py:153: UserWarning: Your
  self.gen.throw(typ, value, traceback)
57/57 ━━━━━━━━━━━━━━━━━━━━ 10s 27ms/step - accuracy: 0.6875 - loss: 0.9689 - val_accuracy: 1.0000 - val_loss: 0.7214
Epoch 3/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 580s 10s/step - accuracy: 0.6381 - loss: 1.0358 - val_accuracy: 0.8033 - val_loss: 0.7482
Epoch 4/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 12s 54ms/step - accuracy: 0.6875 - loss: 0.9111 - val_accuracy: 1.0000 - val_loss: 0.9162
Epoch 5/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 613s 10s/step - accuracy: 0.7104 - loss: 0.8493 - val_accuracy: 0.8217 - val_loss: 0.6451
Epoch 6/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 9s 8ms/step - accuracy: 0.7812 - loss: 0.7638 - val_accuracy: 1.0000 - val_loss: 0.2996
Epoch 7/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 581s 10s/step - accuracy: 0.7176 - loss: 0.7587 - val_accuracy: 0.8162 - val_loss: 0.5916
Epoch 8/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 9s 15ms/step - accuracy: 0.7188 - loss: 0.7600 - val_accuracy: 0.5000 - val_loss: 1.4257
Epoch 9/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 612s 10s/step - accuracy: 0.7464 - loss: 0.6871 - val_accuracy: 0.8199 - val_loss: 0.5476
Epoch 10/10
57/57 ━━━━━━━━━━━━━━━━━━━━ 8s 8ms/step - accuracy: 0.6562 - loss: 0.8759 - val_accuracy: 1.0000 - val_loss: 0.0118
18/18 ━━━━━━━━━━━━━━━━━━━━ 133s 7s/step - accuracy: 0.8138 - loss: 0.5700
Final Test Accuracy: 82.05%
```

Fig3 : output

IV.CONCLUSION

The **BIO-EYE: An Advanced Wildlife Detection Framework** serves as an innovative and essential tool for wildlife conservation, combining cutting-edge artificial intelligence techniques to automate species recognition and habitat monitoring. This system utilizes deep convolutional neural networks (CNNs) for species classification and advanced graphcut algorithms for precise image segmentation, enabling the efficient analysis of wildlife camera trap data. By reducing the need for manual processing, BIO-EYE significantly speeds up the identification process while improving accuracy, ultimately contributing to more effective conservation efforts.The development of BIO-EYE focused on addressing the limitations of traditional wildlife monitoring methods. Manual identification of species from camera trap images is often time-consuming, error-prone, and resource-intensive. BIO-EYE solves this problem by automating the recognition and segmentation process, making it scalable and more accessible for wildlife researchers and conservationists across the globe. The use of CNNs, coupled with transfer learning, allows the system to learn from pre-trained models and adapt to specific wildlife datasets, enhancing its performance in diverse environments. One of the standout features of BIO-EYE is its application of graph-cut algorithms for accurate image segmentation. These algorithms ensure that wildlife is properly isolated from the background, even in complex or cluttered environments. The system is robust enough to handle challenging scenarios such as partial occlusions, varying lighting conditions, and diverse species, which are common in camera trap data. By combining high-quality species classification with precise segmentation, BIO-EYE provides valuable insights into wildlife populations and ecosystems, helping conservationists make informed decisions.The system’s user-friendly interface ensures that it is accessible to a wide range of users, from conservationists to researchers and wildlife enthusiasts. The ability to visualize the detected species and segmented regions enhances the understanding of the data and supports further analysis. Additionally, the potential to scale the system to include more species and handle large datasets makes BIO-EYE a versatile tool that can be used in various geographic regions and ecosystems. Looking toward the future, BIO-EYE offers immense potential for expansion and refinement. One of the key areas for growth is the integration of real-time monitoring capabilities, which would enable automated alerts to conservationists in the event of detected threats such as poaching or environmental disturbances. Furthermore, the system could be enhanced with IoT technology, allowing it to interact with smart cameras and other sensors for more accurate data collection. The application of drones could further extend its reach, enabling wildlife monitoring in remote or inaccessible areas.Another significant direction for future development is the use of AI-driven behavioral analysis, which would allow the system to track patterns in animal movement, social interactions, and population dynamics. Additionally, integrating multimodal data sources,such as acoustic sensors and environmental data, would offer a more comprehensive view of wildlife habitats, enabling deeper analysis of how external factors affect species behavior and distribution. Furthermore, BIO-EYE has the potential to contribute to global biodiversity conservation efforts by sharing data with international databases and supporting collaborative research. As it evolves, it can become a crucial tool in the ongoing battle against biodiversity loss, contributing to habitat restoration and the development of sustainable conservation policies. In conclusion, BIO-EYE is a transformative tool for wildlife detection and conservation, combining advanced AI techniques with practical applications to provide critical support for biodiversity preservation. Its continued development will ensure it plays a key role in tackling the challenges facing wildlife conservation, helping to protect endangered species and preserve ecosystems for future generations.

V.REFERENCES

1. P. M. Vitousek, H. A. Mooney, J. Lubchenco, and J. M. Melillo, “Human domination of Earth’s ecosystems,” Science, vol. 277, no. 5325, pp. 494–499, 1997.

2. G. C. White and R. A. Garrott, Analysis of wildlife radio-tracking data. Elsevier, 2012.

3. R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler, “An analysis of a large-scale habitat

monitoring application," in Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, 2004, pp. 214–226.

4. B. J. Godley, J. Blumenthal, A. Broderick, M. Coyne, M. Godfrey, L. Hawkes, and M. Witt, "Satellite tracking of sea turtles: Where have we been and where do we go next?" Endangered Species Research, vol. 4, no. 1-2, pp. 3–22, 2008.

5. A. Hulbert and J. French, "The accuracy of GPS for wildlife telemetry and habitat mapping," Journal of Applied Ecology, vol. 38, no. 4, pp. 869–878, 2001.

6. R. Kays, S. Tilak, B. Kranstauber, P. A. Jansen, C. Carbone, M. J. Row-cliffe, T. Fountain, J. Eggert, and Z. He, "Monitoring wild animal com- munities with arrays of motion sensitive camera traps," arXiv:1009.5718, 2010.

7. F. O'Connell, J. D. Nichols, and K. U. Karanth, Camera traps in animal ecology: Methods and Analyses. Springer Science & Business Media, 2010.

8. Swanson, M. Kosmala, C. Lintott, R. Simpson, A. Smith, and C. Packer, "Snapshot Serengeti, high-frequency annotated camera trap images of 40 mammalian species in an African savanna," Scientific Data, vol. 2, p. 150026, 2015.

9. S. Thorpe, D. Fize, and C. Marlot, "Speed of processing in the human visual system," Nature, vol. 381, no. 6582, p. 520, 1996.

10. 10. X. Yu, J. Wang, R. Kays, P. A. Jansen, T. Wang, and T. Huang, "Automated identification of animal species in camera trap images, EURASIP Journal on Image and Video Processing, vol. 2013, no. 1, pp. 1–10, 2013.

11. G. Chen, T. X. Han, Z. He, R. Kays, and T. Forrester, "Deep convolutional neural network-based species recognition for wild animal monitoring," in Proceedings of the IEEE International Conference on Image Processing (ICIP), 2014, pp. 858–862.

12. Gómez, A. Salazar, and F. Vargas, "Towards automatic wild animal monitoring: Identification of animal species in camera-trap images using very deep convolutional neural networks," arXiv:1603.06169, 2016.

13. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 248–255.

14. O. Russakovsky, J. Deng, H. Su et al., "ImageNet large scale visual recognition challenge," International

Journal of Computer Vision, vol. 115, no. 3, pp. 211–252, 2015.
15. N. Pinto, D. D. Cox, and J. J. DiCarlo, “Why is real-world visual object recognition hard?” PLOS Computational Biology, vol. 4, no. 1, p. e27, 2008.
16. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural Computation, vol. 1, no. 4, pp. 541–551, 1989.