AniLarm: AI for Small Animal Detection Under Cars Using Thermal Camera

Rahul Choutapally, Konika Reddy Saddikuti and Solomon Berhe
Department of Computer Science, University of the Pacific, USA
Email: r_choutapally@u.pacific.edu, k_saddikuti@u.pacific.edu, sberhe@pacific.edu

***Abstract*—Urban safety concerns, particularly regarding small animals under cars, present a significant challenge. To address this, a research prototype named AniLarm was developed, applying offline AI (no internet dependency) to detect small animals using thermal imaging and machine learning. The system integrates a Seek Thermal Compact Camera with a Raspberry Pi for real-time classification and detection. AniLarm uses the EfficientNet model, outperforming YOLOv5 with a test accuracy of 97.83% on a balanced dataset. The system evolved through multiple iterations, from LED-based alerts to a web application, ultimately adopting an architecture with a Raspberry Pi in a protective case, monitor, and Bluetooth speaker for auditory alerts. Thermal images are processed end-to-end in approximately 8 seconds, with results delivered via audio notifications. The prototype was tested in both controlled environments and real-world scenarios, effectively distinguishing animals from non-animal objects after addressing misclassifications and latency. This research demonstrates the feasibility of integrating an offline trained machine learning model with an embedded system and a vision sensor to improve safety in urban applications.**

***Index Terms*—Animal detection, Thermal Imaging, EfficientNet, Raspberry Pi, Machine Learning, Urban Safety.**

I. Introduction

Urban environments present challenges for humans and animals, particularly stray cats and dogs seeking shelter under parked vehicles during adverse weather. Ratan Tata emphasized the importance of checking under cars to prevent injuries to animals hiding for warmth or protection during monsoons [1]. Cats are instinctively drawn to such spaces due to the warmth and security they offer during cold or rainy weather [2]. However, this poses risks, as animals can be injured or killed when vehicles start, leading to vehicle damage as well [3]. These dangers highlight the need for detection systems to ensure the safety of both animals and vehicles, as drivers may unknowingly start their cars, causing injuries or fatalities to hidden animals (see Fig 1.).

Electric vehicles (EVs) exacerbate this issue due to their near-silent operation compared to combustion engines. The lack of engine noise reduces the chances of animals being alerted to the threat, increasing the risk of harm. As EV adoption grows, this problem is likely to worsen.

Fig. 1. Photo of a cat under a car

Existing solutions for detecting animals under cars are limited and often impractical, with manual checks being time-consuming and difficult in low-light conditions or when drivers are in a hurry. Sensor-based systems may lack accuracy or affordability, and internet-dependent solutions are unreliable in areas with poor connectivity or underground parking.

To address this safety concern, we developed a system, AniLarm, a prototype using thermal imaging and machine learning to detect small animals under cars. It combines a Seek Thermal Compact Camera and a Raspberry Pi 4 Model B to analyze and classify thermal signatures. The core detection uses EfficientNet, a convolutional neural network chosen for its balance between accuracy and computational efficiency. Our prototype underwent several iterations. Initially, a LED-based alert system was considered, but hardware integration challenges led to a web-based application. However, internet connectivity limitations prompted a transition to a completely offline system without requiring internet connectivity. The final iteration features a monitor within a Raspberry Pi case and a Bluetooth speaker, providing audible detection results for safety and accessibility [4].

This paper provides an overview of the system prototype, detailing its development, challenges, and solutions. The paper is structured as follows: Section II reviews related work, Section III discusses the methodology, including data collection, preprocessing, model training, and deployment, Section IV presents experimental results and system evaluations, Section V discusses findings, performance, limitations, and future work, and Section VI summarizes the contributions [5].

By addressing animal safety under cars, our prototype contributes to urban safety and animal welfare, demonstrating the potential of machine learning, embedded systems, and sensors to solve real-world problems.

II. Literature Review

Various animal detection technologies have been developed to improve motor vehicle safety, leveraging methods such as ultrasound, infrared, and thermal detection. Ultrasound operates by analyzing reflected sound waves, while infrared systems detect animals through emitted heat. Thermal detection, on the other hand, visualizes temperature variations, enabling the identification of animals based on their heat signatures. These approaches collectively aim to improve animal detection under different environmental conditions, thereby improving vehicle safety [6]. Complementary to these methods, LiDAR technology has been widely adopted for its ability to create 3D maps of the environment using laser pulses. LiDAR's accurate distance measurement and mapping capabilities significantly contribute to object detection and collision prevention. When combined with cameras, LiDAR improves data acquisition for object tracking and detection efficiency. However, its performance can degrade in adverse weather conditions, such as rain or fog, where measurement accuracy is impacted [7].

Efficient image acquisition is another critical aspect of animal detection systems, providing the foundational data for analysis. Techniques such as motion-triggered camera traps, bio-loggers, drones for inaccessible areas, and satellite imagery for large-scale population studies are commonly used. Each approach has challenges, such as equipment placement, environmental conditions, and the high processing demands of video monitoring, making efficient image acquisition crucial [8]. Thermal imaging cameras, which detect emitted heat rather than visible light, provide reliable vision in conditions where camouflage or darkness obstructs normal sight [9]. The integration of microcontrollers and Wi-Fi communication modules with infrared sensors allows for real-time remote monitoring, making thermal imaging systems highly versatile for both indoor and outdoor animal detection applications [10]. First tested in the 1960s on white-tailed deer (Odocoileus virginianus), thermal cameras have since been used extensively to study and monitor various species, including mammals, birds, and invertebrates [9].

Among image acquisition techniques, camera traps are widely used for wildlife monitoring but often suffer from false triggers, making data processing inefficient. To address this, Tan et al. applied deep learning models such as YOLOv5, FCOS, and Cascade R-CNN for automated animal detection and classification. YOLOv5 excelled, achieving 98.9% mAP, significantly improving processing efficiency for biodiversity research [11]. Popek et al. explored classical (HOG/SVM) and deep learning (YOLOv3, Faster R-CNN) methods for thermal imaging. YOLOv3 proved effective, achieving over 90% mAP at IoU ¿ 50%, but challenges like uneven brightness, overlapping silhouettes, and high computational demands were noted [12]. Integrating neural style transfer with EfficientDet improved thermal object detection, achieving 92.83% mAP and 94.07% accuracy on FLIR test frames. This highlights the impact of advanced preprocessing, such as enhanced resolution and high-frequency detail retention, in optimizing detection for resource-limited environments [13].

Sundaram et al. discussed the application of EfficientNet in animal detection systems, highlighting its efficiency and adaptability. They emphasize its fine-tuning capabilities, which allow it to be tailored to specific datasets for improved performance. The preprocessing steps described include resizing images to a uniform resolution, normalizing pixel values, and applying color mapping, all of which contribute to optimizing the input data for the model [8]. Similarly, Krishnan et al. demonstrated the suitability of EfficientNet for addressing low-quality image challenges in thermal datasets. The model's efficient feature extraction and fine-tuning capabilities enable it to detect cryptic or partially obscured animals effectively. Preprocessing techniques like resizing, normalization, and color mapping were also found to improve its performance [14]. Building on these advancements, channel attention mechanisms, such as those implemented in the YOLOv8-night model, improve model performance by emphasizing key features like animal silhouettes while suppressing background noise. This innovation improves detection reliability across varying animal scales and postures in nighttime environments [15].

In contrast, the work presented here differs from related literature by introducing an offline (no internet connectivity) thermal imaging approach optimized for resource-constrained environments. Unlike models like YOLOv5, this system applies fine-tuned EfficientNet, achieving better accuracy on a balanced dataset. Additionally, it eliminates internet dependency and integrates preprocessing, inference, and auditory feedback into a practical, low-latency architecture, addressing gaps in usability and reliability. The prototype evaluation further demonstrates the system's applicability in real-world use cases. To the best of our knowledge, no previous research has developed or evaluated this combination.

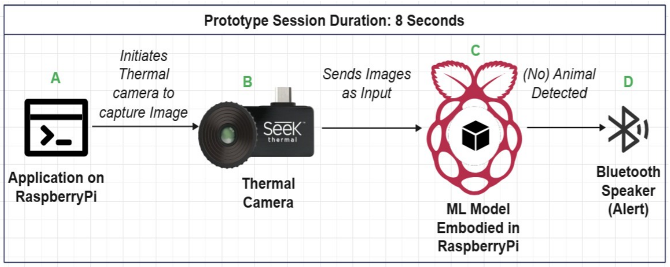

Fig. 2. Architecture Diagram

III. Methodology

A. Architecture

The system integrates hardware and software components towards an offline (no internet dependency) detection functionality. The workflow is structured as follows:

1) Upon triggering the button on application (Fig. 2 A), a thermal image is captured (Fig. 2 B).
2) The image is preprocessed using OpenCV to prepare it for model inference (Fig. 2 C).
3) The EfficientNet model classifies the image into two categories: `Animal` or `No Animal` (Fig. 2 C).
4) The classification result is audibly announced through the Bluetooth speaker, providing immediate feedback (Fig. 2 D).

Initially, it took around 28 seconds from image capture to result announcement because we used seek_viewer to preview the camera feed for 10 seconds and set the capture session to 10 seconds to get more images. Later, we removed both, cutting the time by 20 seconds and bringing it down to just 8 seconds. (see Fig. 2).

B. Data Collection

The early iterations of the system prototype collected thermal images using seek-thermal camera connected to an android mobile device to create a diverse and representative dataset. The dataset consisted of two primary classes:

- **`Animal`:** Thermal images of cats at our university campus were captured using seek-thermal camera as the representative class for animals.
- **`No Animal`:** Initially, we used images of cars and car wheels, but later incorporated human images from a Kaggle dataset [16] to address misclassification issues.

The number of collected images for each class is denoted as N_{animal} and $N_{\text{no\_animal}}$. The datasets for each class are represented as $\mathbf{X}_{\text{animal}}$ and $\mathbf{X}_{\text{no\_animal}}$, respectively:

$$\mathbf{X}_{\text{animal}} = \{x_{ai} \mid x_{ai} \in \mathbf{X}_{\text{animal}}, i = 0, 1, \ldots, N_{\text{animal}}\} \quad (1)$$

$$\mathbf{X}_{\text{no\_animal}} = \{x_{ni} \mid x_{ni} \in \mathbf{X}_{\text{no\_animal}}, i = 0, 1, \ldots, N_{\text{no\_animal}}\} \quad (2)$$

Lastly, both datasets were combined into a single dataset $\mathbf{X}$, labeled appropriately as `Animal` or `No Animal`:

$$\mathbf{X} = \{\mathbf{X}_{\text{animal}}, \mathbf{X}_{\text{no\_animal}}\} \quad (3)$$

The dataset was then divided into three subsets: training ($\mathbf{X}_{\text{train}}$), validation ($\mathbf{X}_{\text{val}}$), and testing ($\mathbf{X}_{\text{test}}$). The split ratio was 70:20:10 among 1380 images in total, ensuring sufficient data for each phase:

$$\mathbf{X} = \mathbf{X}_{\text{train}} \cup \mathbf{X}_{\text{val}} \cup \mathbf{X}_{\text{test}} \quad (4)$$

C. Data Preprocessing

After collecting the data, preprocessing was performed to prepare it for model training. Key preprocessing steps included:

1) **Resizing:** All images were resized to 224×224 pixels to match the input size required by the EfficientNet model.
2) **Normalization:** Pixel values were scaled to the range [-1, 1] to align with the model's training configuration.
3) **Color Mapping:** Grayscale thermal images were converted to color-mapped representations using OpenCV to highlight heat signatures.

For a given thermal image Img_i, preprocessing operations were represented as:

$$\text{Img}_{\text{processed}} = \text{ColorMap}(\text{Normalize}(\text{Resize}(\text{Img}_i))) \quad (5)$$

D. Model Training

The EfficientNetB0 model, pre-trained on the ImageNet dataset, was used as the base for this task. This pre-trained model served as an effective feature extractor, applying its prior training on a large-scale dataset. To adapt it to the specific problem of detecting small animals in thermal images, we manually fine-tuned the model using our custom dataset. During the initial training phase, the base model was frozen, retaining its general feature extraction capabilities and reducing the computational load. Additional trainable layers were added on top of the base model to tailor the network to the specific dataset. Once these layers were trained, the entire model was unfrozen, and the deeper layers of the base model were fine-tuned using the training dataset $\mathbf{X}_{\text{train}}$. This phased fine-tuning approach ensured optimal learning while minimizing the risk of overfitting.

The training process aimed to optimize classification accuracy and reduce loss. Let the machine learning classifier be represented by $\mathcal{M}$. The training phase can be expressed as:

$$\mathcal{M}_{\text{trained}} = \mathcal{M}(\mathbf{X}_{\text{train}}) \quad (6)$$

After training, the model was validated on $\mathbf{X}_{\text{val}}$, which helped in fine-tuning hyperparameters such as learning rate and regularization. Finally, the trained model was evaluated on the test set $\mathbf{X}_{\text{test}}$, achieving a high accuracy of 97.83%. The manual fine-tuning of the EfficientNetB0 model significantly improved its ability to distinguish between `Animal` and `No Animal` classes in thermal image data, addressing the unique challenges of this application.

E. Model Deployment

The trained and fine-tuned EfficientNet model was deployed on a Raspberry Pi for real-time inference. Initial attempts at TensorFlow Lite conversion reduced model accuracy, prompting the direct deployment of the TensorFlow `.h5` model. This deployment process included:

1) Exporting the model in the TensorFlow SavedModel format.
2) Integrating the model with a Python-based application to process images, generate results and display them on the application.
3) Connecting the application to a Bluetooth speaker for auditory alerts, ensuring usability in offline (no internet connectivity) environments.

The deployed model is represented as $\mathcal{M}_{\text{deployed}}$, processing input data $\mathbf{X}_{\text{test}}$ as:

$$\mathbf{Y}_{\text{predicted}} = \mathcal{M}_{\text{deployed}}(\mathbf{X}_{\text{test}}) \tag{7}$$

Where $\mathbf{Y}_{\text{predicted}}$ represents the detection results (`Animal` or `No Animal`) for the test images.

IV. Results

A. Hardware Setup

The system prototype underwent several iterations of hardware adaptations to address integration challenges. Initially, an LED setup was considered to provide visual alerts; however, physical component integration posed a limitation. Subsequently, a web-based application was explored, but the reliance on internet connectivity proved to be another constraint. Eventually, the system evolved to a complete offline system, with a Raspberry Pi connected to a monitor housed within a Raspberry Pi case and a Bluetooth speaker for auditory notifications. This setup ensured accessibility and reliability while providing real-time results.

The Seek Thermal Compact Camera was used to capture thermal images under cars. The Raspberry Pi 4 Model B served as the processing unit, capable of running machine learning models and interfacing with the peripherals for output. The Bluetooth speaker is connected to raspberry pi supporting bluetooth version 4.0 and above to announce detection results audibly, further improving the system's usability. This complete hardware configuration allowed the prototype to operate effectively in various environments, including urban areas with limited infrastructure.

B. Software Setup

The software stack for AniLarm was developed to maximize compatibility and performance. Python served as the primary programming language, with TensorFlow/Keras utilized for model training and inference, and OpenCV for preprocessing thermal images. A virtual environment was configured on the Raspberry Pi to manage dependencies and ensure a lightweight runtime.

The EfficientNet model, fine-tuned during the training phase, was deployed in its TensorFlow format (`.h5`) for real-time detection. Although TensorFlow Lite was initially considered for deployment, its reduced accuracy necessitated the use of the full TensorFlow model to maintain high classification performance. The integration of the model with the Python-based application ensured seamless functionality, including preprocessing, inference, and output generation.

C. Evaluation Criteria

To evaluate the performance of AniLarm, the following criteria were established:

- **Accuracy:** The percentage of correctly classified images in the test dataset.
- **Precision:** The proportion of true positive detections among all positive detections, representing the model's ability to avoid false positives.
- **Recall:** The proportion of true positive detections among all actual positives, representing the model's sensitivity.
- **F1-Score:** The harmonic mean of precision and recall, providing a balanced measure of the model's performance.
- **Latency:** The total time taken by the system to process an image and generate results, measured in seconds.

These metrics ensured a comprehensive evaluation of the system's effectiveness, balancing accuracy, sensitivity, and usability.

D. Experimentation and Analysis

The AniLarm system was tested on a dataset of 276 thermal images, equally divided into `Animal` and `No Animal` categories. A comparative analysis of YOLOv5 and EfficientNet models was conducted to evaluate their performance in detecting small animals under vehicles.

For the `No Animal Detected` class, the YOLOv5 model achieved a recall of 0.95 but had low precision (0.48), resulting in an F1-score of 0.64. For the `Animal Detected` class, YOLOv5 struggled significantly, with a precision of 0.53, a recall of 0.06, and an F1-score of 0.10. The overall accuracy of YOLOv5 was 48%, with macro and weighted average F1-scores of 0.37 and 0.36, respectively, reflecting its limitations in balancing precision and recall. The classification

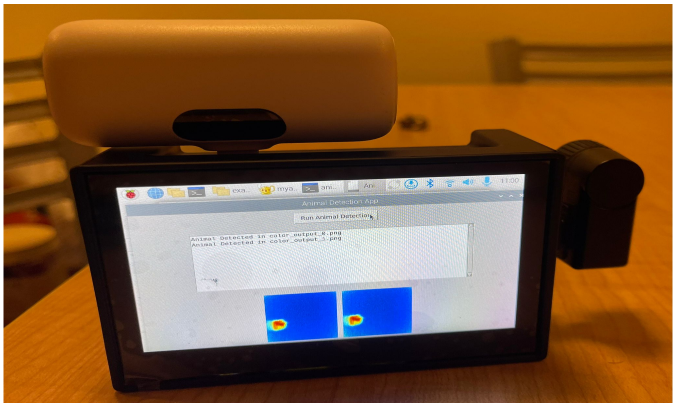

Fig. 3. Animal Detected

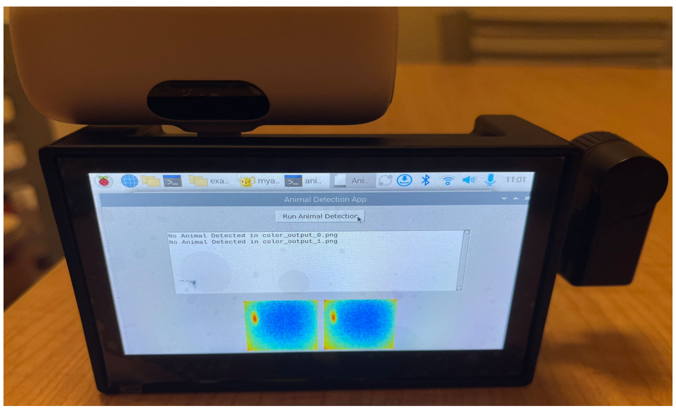

Fig. 4. No Animal Detected

report for YOLOv5 is detailed in Table I. In contrast, the

TABLE I
CLASSIFICATION REPORT FOR YOLOV5 MODEL

Class	**Precision**	**Recall**	**F1-Score**	**Support**
No Animal	0.48	0.95	0.64	132
Animal	0.53	0.06	0.10	144
Accuracy		0.48 (276)		
Macro Avg	0.51	0.50	0.37	276
Weighted Avg	0.51	0.48	0.36	276

EfficientNet model demonstrated superior performance across all metrics. For the `Animal` class, it achieved a precision of 1.00, a recall of 0.95, and an F1-score of 0.98. Similarly, for the `No Animal` class, EfficientNet achieved a precision of 0.96, a recall of 1.00, and an F1-score of 0.98. This balanced performance resulted in an overall accuracy of 97.83%, with macro and weighted average F1-scores of 0.98. The classification report for EfficientNet is presented in Table II, and its accuracy progression during training is shown in Figure 5.

TABLE II
CLASSIFICATION REPORT FOR EFFICIENTNET MODEL

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Animal	1.00	0.95	0.98	132
No Animal	0.96	1.00	0.98	144
Accuracy		0.9783 (276)		
Macro Avg	0.98	0.98	0.98	276
Weighted Avg	0.98	0.98	0.98	276

EfficientNet not only achieved better accuracy than YOLOv5 but also demonstrated superior compatibility with the Raspberry Pi's computational constraints. While YOLOv5 struggled with the processing requirements and nuances of thermal image classification, EfficientNet provided a more efficient and reliable solution for real-time deployment. These results validated EfficientNet as the optimal model for AniLarm, ensuring high accuracy and reliability in offline (no internet connectivity) environments.

In conclusion, the comparative analysis highlights EfficientNet's ability to outperform YOLOv5 across all key metrics, making it the preferred model for AniLarm's deployment. This experimentation and analysis phase underscored the importance of selecting an efficient and robust model for small animal detection using thermal imaging.

E. Latency Analysis:

Initially, it took around 28 seconds from image capture to result announcement because we used seek_viewer to preview the camera feed for 10 seconds and set the capture session to 10 seconds to get more images (about 2 images). Later, we removed both, cutting the time by 20 seconds and bringing it down to just 8 seconds. This latency includes image capture, preprocessing, inference, and result generation. Although this is suitable to reduce the latency operation, future iterations will focus on reducing it by using different model architectures and faster hardware.

F. Prototype Performance:

- **Animal Detected:** The system successfully identified a heated brass cat as an animal, demonstrating its ability to process and classify thermal images accurately as shown in figure3.
- **No Animal Detected:** The system correctly classified empty frames without any false positives, ensuring robustness in non-detection scenarios as shown in figure4.

V. DISCUSSION

AniLarm demonstrated the potential of thermal imaging and machine learning for urban safety, achieving 97.83% test

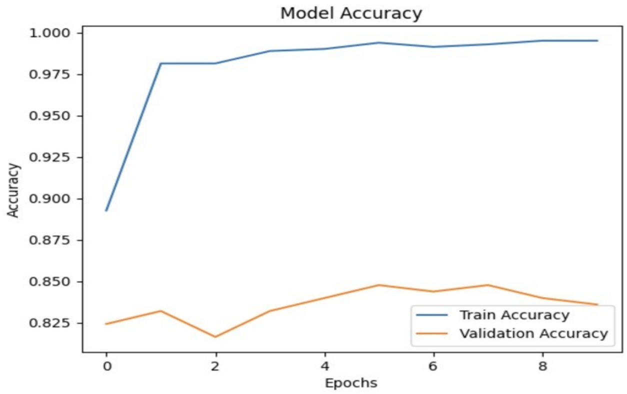

Fig. 5. Graph for EfficientNet Model Accuracy

accuracy and reliable performance in offline environments (without internet access). Using EfficientNet, it outperformed YOLOv5 in accuracy, resource efficiency, and Raspberry Pi compatibility. However, limitations included a dataset focused on cats and certain human poses, restricting generalization, and a single thermal camera limiting the detection field of view. The 8-second processing time also introduced latency, while hardware constraints prevented more advanced optimizations. These challenges were mitigated through dataset augmentation, model refinement, and careful tuning. Future improvements should expand the dataset, conduct field trials, and develop adaptive algorithms to address environmental factors such as varying temperatures or reflective surfaces. Multi-sensor integration and upgraded hardware, such as NVIDIA Jetson Nano, could further enhance detection and reduce latency. The AniLarm architecture is adaptable for applications such as pedestrian safety, wildlife monitoring, and industrial safety systems, demonstrating its adaptability and scalability. Overall, AniLarm successfully addresses a critical urban safety challenge while providing a solid foundation for future improvements and broader applications.

VI. Conclusion

In this research study, our prototype AniLarm system addresses urban safety by detecting small animals under vehicles using thermal imaging and machine learning. Achieving 97.83% accuracy with a custom-trained EfficientNet model, it outperforms YOLOv5 in both accuracy and efficiency. The offline design, using a Raspberry Pi and Seek Thermal Camera, ensures reliable operation in resource-constrained environments. Challenges such as latency and hardware limitations were addressed through dataset augmentation and model refinement. While the system's reliance on thermal imaging and a single camera limits its scalability, future work can improve its robustness by expanding the dataset, integrating multiple sensors, and optimizing processing times. AniLarm's adaptable architecture also holds promise for broader applications in wildlife monitoring and safety systems. This research demonstrates the potential of machine learning and embedded systems in solving real-world safety challenges.

References

[1] "What ratan tata said about stray animals' safety during monsoon - news18." https://www.news18.com/business/what-ratan-tata-said-about-stray-animals-safety-during-monsoon-8255647.html. [Accessed 08-09-2024].

[2] "Where do cats hide under cars?, 21cats.org." https://www.21cats.org/where-do-cats-hide-under-cars/. [Accessed 08-09-2024].

[3] "How to keep cats from under your car - vet explains pets." https://vetexplainspets.com/how-to-keep-cats-from-under-your-car/. [Accessed 08-09-2024].

[4] R. Sedouram and D. V. Klein, "Safer audio consumption while driving." Technical Disclosure Commons, April 02 2023.

[5] R. Choutapally, K. R. Saddikuti, and S. Berhe, "Poster: Securing iot edge devices: Applying nist ir 8259a to a realtime animal detection system." Workshop on Security and Privacy in Standardized IoT (SDIoTSec), 2025.

[6] V. Sibanda, K. Mpofu, J. Trimble, and N. Zengeni, "Design of an animal detection system for motor vehicle drivers," *Procedia CIRP*, vol. 84, pp. 755–760, 2019. 29th CIRP Design Conference, 08-10 May 2019, Póvoa de Varzim, Portugal.

[7] Y. Waykar, "Lidar technology: A comprehensive review and future prospects," *International Journal of Emerging Technologies and Innovative Research*, vol. 9, no. 7, pp. h164–h172, 2022. Available at www.jetir.org.

[8] N. Sundaram and S. D. Meena, "Integrated animal monitoring system with animal detection and classification capabilities: A review on image modality, techniques, applications, and challenges," *Artificial Intelligence Review*, vol. 56, no. 1, pp. 1–51, 2023.

[9] D. Karp, "Detecting small and cryptic animals by combining thermography and a wildlife detection dog," *Scientific Reports*, vol. 10, no. 1, p. 5220, 2020.

[10] Z. Wang and X. Liu, "Design of animal detector based on thermal imaging sensor," in *Journal of Physics: Conference Series*, vol. 1550, p. 042066, IOP Publishing, 2020.

[11] M. Tan, W. Chao, J.-K. Cheng, M. Zhou, Y. Ma, X. Jiang, J. Ge, L. Yu, and L. Feng, "Animal detection and classification from camera trap images using different mainstream object detection architectures," *Animals*, vol. 12, no. 15, 2022.

[12] L. Popek, R. Perz, and G. Galiski, "Comparison of different methods of animal detection and recognition on thermal camera images," *Electronics*, vol. 12, no. 2, p. 270, 2023.

[13] A. Gautam and S. Singh, "Neural style transfer combined with efficientdet for thermal surveillance," *The Visual Computer*, vol. 38, no. 12, pp. 4111–4127, 2022.

[14] B. S. Krishnan, L. R. Jones, J. A. Elmore, S. Samiappan, K. O. Evans, M. B. Pfeiffer, B. F. Blackwell, and R. B. Iglay, "Fusion of visible and thermal images improves automated detection and classification of animals for drone surveys," *Scientific Reports*, vol. 13, no. 1, p. 10385, 2023.

[15] T. Wang, S. Ren, and H. Zhang, "Nighttime wildlife object detection based on yolov8-night," *Electronics Letters*, vol. 60, no. 15, p. e13305, 2024.

[16] "Thermal image dataset for object classification." https://www.kaggle.com/datasets/breejeshdhar/thermal-image-dataset-for-object-classification. [Accessed 17-11-2024].