Multi-agent AI system for Wildlife detection and traffic accident

Tan Thuy Linh Dao[1], Buu Quan Luu[2], and Tung Duong Nguyen[3]

[1]The University of Newcastle Australia

CORRESPONDING AUTHOR: Tan Thuy Linh Dao (e-mail: c3425175@uon.edu.au).

The authors contributed equally to this article. This work was supported by the University of Newcastle Australia.

ABSTRACT Wildlife-vehicle collisions (WVCs) are a growing concern in forest-adjacent road and railway areas, particularly in countries like Australia and the United States. These incidents not only threaten ecological systems and biodiversity but also pose serious risks to human safety and cause significant economic losses. This research aims to develop an AI-powered multi-agent system for real-time wildlife detection and proactive driver alerting. The system integrates a pretrained YOLOv8x deep learning model for animal detection with a coordinated multi-agent architecture, including Perception, Threat Assessment, Communication, and Monitoring agents. Together, these agents facilitate rapid detection, risk evaluation, real-time alerts, and event logging. The model's performance is evaluated using metrics such as mean Average Precision (mAP), precision, and recall, and is benchmarked against other YOLO variants and a Faster R-CNN model. The proposed system contributes to Sustainable Development Goal 11 (Sustainable Cities and Communities) by supporting safer, more resilient transport networks and protecting biodiversity in urban-fringe areas. The results demonstrate that the system not only achieves high detection accuracy but also enables scalable deployment and timely interventions that can reduce wildlife-vehicle accidents in critical regions

IEEE SOCIETY/COUNCIL Power and Energy Society (PES)

DATA DOI/PID <Will add later>

DATA TYPE/LOCATION Images, Callaghan Campus, Newcastle, NSW

INDEX TERMS Artificial Intelligence, Machine Learning, Detection, Deep Learning, Multi-agent, Wildlife, YOLO, Wildlife-Human Collisions (WHCs), You Only Look One

INTRODUCTION

Road accidents between animals and vehicles are a serious problem in many countries with large wildlife populations, such as Australia and the United States, growing threat to both ecological systems, human safety, and substantial economic. Several factors contribute to the rate of collisions, including vehicle speed, animal behaviour, roadside vegetation, driver awareness, and traffic volume. As Litvaitis and Tash (2008) highlight, the interplay of landscape integrity, animal abundance, and habitat proximity to roads significantly influences collision risk, making the problem both spatially and temporally dynamic [1]. This can be seen in the state of Victoria, Australia, during the 10-year period from 2007 to 2016, there is an increase in 6.7% per year in the rate of accidents between the animals and vehicles in the road, which highlights the urgency of developing more real-time solutions to prevent these situations [2].

Specifically, the dynamic and AI-powered systems that can detect wildlife near roads, assess the rate of potential collisions, and provide timely alerts to drivers. The systems align with the United Nations Sustainable Development Goal 11 (SDG 11), which advocates for sustainable and safe transport infrastructure bridging the gap between urban development and wildlife conservation [3]. The design and implementation of a multi-agent AI system brings together a cohesive team of specialized agents, including Perception, Threat Assessment, Communication, and Monitoring. This sophisticated system is thoughtfully crafted to consider factors such as species type, proximity to the roadway, and the critical need for timely alerts. By expertly managing decision-making processes, it paves the way for effective and impactful implementation, ensuring a safer and more responsive environment for all [4]. At the core of the Perception Agent is the YOLO v8 deep learning model, a popular object detection architecture which known as fast and accurate method for animals' detection. The model will be trained on the large set of animals' images and then detect the presence of animals in an image by bounding boxes and class probability class [FIGURE 1].

This report outlines the growth and evaluation of an AI-powered wildlife detection to avoid the traffic accidents between animals and humans. The structure is designed to guide through the conceptual design, technical implementation, its performance, and future applications, which will be organized in the following sections.

LITERATURE REVIEW

1. Key findings

Reichstein et al. (2019) provide a comprehensive analysis of how Artificial Intelligence, particularly Machine Learning, is increasingly applied to address difficult challenges in ecology and Earth system science. The authors emphasize that environmental systems are inherently multifaceted, involving dynamic interactions among variables such as climate patterns, land usage, and animal behaviour. Traditional analytical methods often struggle to capture the nonlinear relationships and vast data scales involved in such systems. However, Machine Learning offers a powerful alternative, capable of processing and extracting valuable insights from large-scale datasets gathered by satellites, environmental sensors, and wildlife monitoring cameras. This allows for more data-driven and adaptive approaches to understanding and controlling ecological systems. [5]

The use of deep learning, more especially deep neural networks (DNNs), for the automatic identification, counting, and description of wild animals caught in camera-trap photos is examined by Norouzzadeh et al. (2018). Utilizing the Snapshot Serengeti (SS) dataset, which is one of the most extensive camera-trap datasets globally, comprising over 1.2 million capture events from 255 camera traps—the study assesses the performance of state-of-the-art computer vision models in wildlife monitoring. The results demonstrate that DNNs can achieve high accuracy in species classification and behavioural annotation tasks, highlighting their potential to significantly enhance ecological research by automating traditionally labour-intensive image analysis processes. [1]

To increase traveller safety in forest corridors, Raj et al. (2023) develop an AI-driven system for identifying wild animal invasions. The system makes use of an Internet of Things (IoT) framework for remote monitoring and a Raspberry Pi 3 Model B for real-time image capture. It employs the YOLO (You Only Look Once) object detection algorithm, alongside libraries such as NumPy and OpenCV for image processing. The model is trained on a large, annotated dataset of animal images, utilizing TensorFlow as the core deep learning framework. This efficient and cost-effective solution effectively detects animal presence in sensitive areas, reducing the risk of wildlife-vehicle collisions. [4]

Senthil et al. (2024) propose Safe Road AI, a real-time accident detection system utilizing deep learning and computer vision to analyse multi-angle crash video footage. The model is based on a convolutional neural network (CNN) architecture trained using a labelled dataset obtained from Kaggle. To ensure efficient processing, the input images are pre-processed by resizing them to a standardized resolution, facilitating seamless integration into the CNN framework. The system leverages supervised learning techniques and is designed to work with IoT-enabled camera feeds, enabling real-time detection of road accidents. This approach highlights the potential of CNN-based architectures in enhancing roadway safety through intelligent video analysis and rapid event detection. [6]

A deep learning-based obstacle detection and categorisation system is presented by Prabhakar et al. (2017) with the goal of enhancing safety in situations involving high-speed autonomous driving. The approach employs the Faster R-CNN algorithm to process real-time video input from cameras at a rate of 10 frames per second using GPU acceleration. The system is capable of accurately identifying various road entities, including vehicles, pedestrians, and animals, under diverse environmental conditions. Particularly, it achieves a mean average precision (mAP) of 97.42 percent on Indian roads, outperforming conventional sensor-based systems such as LiDAR in terms of cost-effectiveness and adaptability. Furthermore, the model demonstrates a high degree of robustness to weather and lighting fluctuations, which are critical for ensuring reliable and safe autonomous navigation. [7]

This study introduces a YOLOv5-based system designed for nighttime animal detection on highways, utilizing CLAHE and Retinex preprocessing techniques to enhance low light images (Parkavi et al., 2025). The dataset consists of 927 images, which is divided 854 for training, 38 for validation, and 35 for testing sourced from Roboflow, featuring deer, foxes, and bears in highway and forest environments. The preprocessingphase combines CLAHE for local contrast enhancement with Retinex for illumination normalization, significantly improving the input quality for YOLOv5. The model achieves impressive performance metrics, with 92.3 percent precision and 77.3 percent recall, while maintaining real-time functionality at 55 – 60 FPS. Despite its strong performance in typical darkness, its accuracy declines in adverse weather conditions such as rain and fog, attributed to the limited data available for these scenarios. Future efforts will focus on expanding the dataset and optimizing the model for edge deployment. This AI-driven solution aims to enhance road safety by mitigating animal-vehicle collisions. [8]

2. Critical Evaluation

The reviewed literature reveals AI's numerous applications in animal conservation and road safety, each with unique characteristics. Reichstein et al. (2019) emphasise AI's broad potential in Earth system science, while Norouzzadeh et al. (2018) focus specifically on automating species recognition in camera-trap images using DNNs, achieving excellent accuracy on the Snapshot Serengeti dataset. Raj et al. (2023) adopt a lightweight, IoT-based approach with YOLO on Raspberry Pi for real-time wildlife detection in forest corridors, highlighting deployability over computational power. In contrast, Prabhakar et al. (2017) leverage GPU-accelerated Faster R-CNN for high-speed autonomous driving, excelling in obstacle detection (97.42% mAP) but requires more resources. Parkavi et al. (2025) focus on nighttime animal detection using YOLOv5 with CLAHE/Retinex pre-processing, achieving 92.3% precision but struggling in adverse weather, whereas Senthil et al. (2024) create a broader CNN-based system for multi-angle crash detection, emphasizing real-time accident response. Collectively, these efforts demonstrate trade-offs between scope (ecological vs. road safety), computational demands (edge vs. GPU), and environmental adaptability, underscoring the importance of robust, scalable solutions in future study.

BACKGROUND - TECHNICAL FOUNDATION

1. Problem context

Wildlife-vehicle collisions are one of the most significant impacts of roads and railways globally. The rate of morality is considered to be startling – more than 350 million vertebrates are killed annually in the US and around 194 million birds and 29 million mammals are killed across Europe each year. Within Australia, Approximate four million marsupials and six million birds, plus other fauna groups, are killed annually. WVC cause significant damage for both people and wildlife. For humans, they can injury or death, as well as significant property damage to the vehicle involved. For wildlife, they can disrupt natural migration patterns and even contribute to declining populations of certain species. Specifically, in Australia, analysis of more than 21,000 AAMI animal collision claims across the country in 2023 found [9] that - NSW is the most dangerous state for animal collisions (30 percent) followed by Vic (29 percent) and Queensland (24 percent)
- 36 percent of animal collisions occur on rural and regional roads

2. Relevant to SDGs

SDG11 seeks to address several interrelated issues by focusing on urban sustainability. Like other SDG goals and targets, it is based on the systems approach with an emphasis on cross-linkages with other developmental priorities and aims to achieve the desired level of outcome by 2030. The concept of 'sustainable development' rests on achieving balance between economic, social and environmental objectives. The adaption of the Sustainable Development Goals including the stand-alone urban goal of making cities safe, inclusive, resilient and sustainable firmly places urbanisation at the forefront of international development policy. Thus Goal 11 is most relevant and seeks to make cities and human settlement inclusive, safe, resilient and sustainable through eliminating slum like conditions, providing accessible and affordable transport systems, reducing urban sprawl, increasing participation in urban governance, enhancing cultural and heritage preservation, addressing urban resilience and climate change challenges, better management of urban environments (pollution and waste management), providing access to safe and secure public spaces for all [10].

These collisions disrupt both ecological and human communities, directly undermines progress toward SDG11 by compromising safe, sustainable transport systems and urban environmental resilience. High WVC rates increase road fatalities, infrastructure damage, and habitat fragmentation againsting SDG11's aim to reduce environmental impacts and fostering inclusive communities. Accident hot spots increase disparities in access to safe roads by interacting with connectivity for rural and urban communities. This project directly addresses these challenges by deploying an AI-powered wildlife detection system that enhances road safety while supporting sustainable infrastructure development. By providing real-time collision warnings, the system aligns with SDG 11's call for safer transport networks, while its minimal ecological disruption supports SDG 11's goal of protecting natural heritage. Furthermore, the technology's scalability offers a model for innovative, sustainable urban planning, demonstrating how smart systems can reconcile human mobility with biodiversity conservation, which is a critical step toward achieving SDG 11's vision of resilient, future-ready cities.

3. Deep Learning YOLOv8x

The primary role of the YOLO target detection algorithm lies in the model's small size and fast calculation speed. The structure of YOLO is straightforward, allowing it to directly output both the position and category of bounding boxes through a neural network. The speed of YOLO is fast because YOLO only needs to put the picture into the network to get the final detection result, enabling YOLO can also realise the time detection of video [11]. The original YOLO design has 24 convolutional layers and two fully connected layers. YOLO predict multiple bounding boxes per grid cell; however, it only keeps bounding boxes having highest Intersection Over Union (IOU) with the ground truth known as non-maximum suppression [12].

To eliminate these mentioned risks and enhance SDG 11, advanced detection solutions such as YOLOv8x (You Only Look One version 8 extra-large) have utilised as a powerful tool for real-time wildlife detection in transportation corridors. This is the most robust model in the YOLOv8 series developed by Ultralytics built on the deep learning-based object detection framework using Convolutional Neural Networks (CNNs) to excels in identifying and localizing objects. The model's anchor-free architecture, combined with its decoupled head design and efficient implementation in PyTorch, makes it exceptionally well-suited for dynamic environments [13].

The integration of YOLOv8X into roadside detection systems shows a critical advancement in aligning transportation safety with biodiversity conservation. By enabling real-time detection and alerting of wildlife near roads, this technology not only avoids accidents but also supports sustainable infrastructure development with minimal ecological disruption. By encouraging safer travel and conserving natural environment, it advances SDG 11 and shows how intelligent systems can strengthen environmental and human communities' resilience. This research will employ pre-trained YOLOv8x model to perform the wildlife detection component of the system. The Ultralytics YOLO v8 library was imported using the Python, which is a comprehensive open source providing the pre-trained models for object detection, classification, and segmentation. This offer APIs for the purposes of training, validation, inference and exporting the dataset.

METHODOLOGY

Our AI-powered wildlife detection system employs a collaborative multi-agent framework for real-time protection. The process is started by the PerceptionAgent using YOLO v8x TO identify the species and location. After that, The ThreatAssessmentAgent evaluates risk levels based on factors such as proximity to roads and time of day, while the CommunicationAgent initiaties roadside warnings to warn drivers in high and medium risk cases. Together, these agents form a closed-loop pipeline that operates in milliseconds—from detection to decision to action—ensuring both wildlife protection and road safety with minimal false alarms. [Figure 2,3,4]

1. Dataset

The Animal Object Detection dataset hosted on Roboflow Universe by Tran Tien Van, is a comprehensive resource for training object detection and facilitating the development of computer vision models to recognise the most popular two specific Australian animals: kangaroos and koalas. This dataset includes 2655 open-source images divided into the two classifications indicated above. Each image is labelled to identify instances of kangaroos and koalas, along with bounding boxes that reflect the location of each animal within the image. Furthermore, the dataset is released under the CC BY 4.0 license allowing for widespread usage and adaptation with appropriate attribution.

To ensure robust model performance, the dataset was throroughly prepared. First, label files were standardised by correcting class indices (0 for Kangaroo, 1 for Koala) to correspond to the model's preset classes. A structured data.yaml file was created, including class names, counts, and accurate paths to training/ validation directories. The dataset was reorganised into YOLO-compliant /images and /labels subfolders. Data augmentation techniques such as rotation, blurring, and brightness modifications were applied to increase diversity, while mosaic augmentation (2x2 composites) and CutMix (partial image fusion) improved the model's cabability to handle occlusions and complicated scenes. Finally, manual verification guranteed every image had a correctly formatted label,which eliminated path errors or mismatches.

The dataset is spitted into three main subsets to support the development and evaluation of the object detection models. For the training sets, it accounts for 82 percent of the images are allocated for training purposes allowing the model to learn to identify and distinguish between the two animals. To help adjust model parameters and prevent overfitting during training, it is necessary to set 12 percent of the images for the validation. The remaining 6% of the images are reserved for testing which provides an unbiased evaluation of the model's performance on previously unseen images

2. Constraints

w

2.1. Real-Time Processing Requirements:

The system must process each camera frame and complete the full agent pipeline (detection, risk assessment, and communication) in less than 100 milliseconds to ensure timely driver alerts

2.2. Detection Performance Trade-offs:

The system must strike a delicate balance between detection sensitivity (recall) and precision to ensure both road safety and user trust. High sensitivity (recall) is prioritised to minimise false negatives (FN), as missing an animal that poses a threat could result in collisions. However, excessive sensitivity may raise false positives (FP), resulting in unnecessary driver alerts and lowering system confidence. To address this, the ThreatAssessmentAgent dynamically modifies risk scoring based on per-species model sensitivity, lowering the impact of low-confidence detections when recall is inherently weaker for certain animals. Achieving an optimal trade-off is essential. High recall enhances safety by ensuring the system detects the majority of actual animal threats, lowering the danger of accidents; however, this increased sensitivity may lead to frequent alerts that could overwhelm drivers and reduce their reliance on the system. On the other hand, high precision decreases false alarms by only sending warnings for confident detections, enhancing user confidence, but this stricter filtering risks missing certain animals—particularly in tough conditions such as limited visibility, obstructed views, or fast-moving scenarios.

2.3. Power Consumption Limits:

To operate sustainability in off-grid conditions utilising solar or battery power, the system's power consumption must be minimised. This requires low-power hardware and efficiency optimisations like model pruning, quantisation, and edge TPU acceleration for the Perception Agent. These techniques eliminate computational demands while maintaining detection accuracy. Intelligent power management, which includes adaptive sleep modes and processing rates, further extends operational time by conserving energy during low-activity periods. Such optimizations ensure reliable, long-term deployment in remote areas without compromising safety performance.

2.4. Environmental Conditions (Weather/ Lighting):

The accuracy of the system might be affected by tough weather and conditions that impact detection reliability. For example, rain, fog, and snow cannot detect object properly, while low-light situations during night, dusk, or dawn periods may compromise detection accuracy. For the system to be effective, it must have robust design features including enhanced perception capabilities, such as thermal or infrared imaging support in the PerceptionAgent, and adaptive threat evaluation, for instance time-weighted scoring mechanisms in the ThreatAssessmentAgent. The design of the solution should account for these features, and it must do the same during deployment so that wildlife can be detected safely in any weather or under any light.

2.5. Occlusion and Partial Visibility

The system must manage significant challenges when animals cannot be easily spotted by vegetation, road obstacles or other vehicles nearby. The PerceptionAgent should perform temporal analysis and track moving objects to keep track of the same person even if they go out of view, so identification is not disrupted between frames. Simultaneously, the ThreatAssessmentAgent must implement probabilities to uncertain detections of obscured animals, balancing caution against over-alerting and reduce false alerts. This dual strategy ensures the system remains effective when confronted with real-world visibility limitations while guaranteeing the system continues to function well in the real-world context visibility constraints. The approach requires careful optimisation of both detection persistence levels and risk evaluation algorithms.

3. Camera

The Raspberry Pi deployment camera serves as the primary input source in the wildlife detection system. Its fundamental purpose is to capture real-time image or video frames in outdoor environments. The Raspberry Pi Camera Module v3 is a compact, high-resolution camera featuring a 12MP Sony IMX708 sensor with HDR, autofocus, and low-light support. It records up to 1080p at 50fps or 720p at 100fps, with standard and wide-angle lens options, making it ideal for outdoor wildlife monitoring in various lighting conditions. Furthermore, it is considered as an affordable security systems, offering the better connectivity between transferring and receiving through computer network and the internet [14] HDR on the camera significantly improves visibility in environments with dynamic lighting, such as forest corridors with shadows or sunlit roads, enabling animals are presented well on images which in turn improves how accurate the bounding boxes are. Additionally, the wide field of view (FOV) supports the detection of multiple animals within a single frame, making each inference more efficient spatially. The high frame rate options help reduce motion blur when monitoring fast-moving species like kangaroos, while the low latency input ensures that frames are fed into the Perception Agent in real-time with minimal delay. Beyond its technical strengths, the camera's affordability and compatibility with low-power Raspberry Pi units make it a sustainable and scalable solution for large-area deployment. Moreover, the animal detection solution allows real-time and budget-friendly spotting of crossing animals which protects people, helps prevent accidents on highways and maintains a natural balance in nearby areas, supporting SDG 11.

4. Multi-Agent

4.1. Perception Agent

4.1.1. Training YOLOv8x model

[Figure 3] To create the final prediction in YOLO, we only need to look at the network once and make one forward pass across it. The system for object prediction is made up of a convolutional neural network [FIGURE 5]. In addition to class labels, the object prediction algorithm predicts item location and shape. Tensor flow (deep learning) is utilised in the numerical computation and image processing tools YOLO, NumPy, and openCV. Finally, import the display() Python function to show the image. The architecture receives the training set as an input in the form of image represented in Figure 3. The architecture then resized the photos to 640x640 as the model was evaluated at two different input image resolutions: 320 x 320 and 640 x 640 [5] while preserving the original AR and padding

Later, CNN broadcasts the image. Following two completely linked layers, there are 24 convolution layers and 4mx pooling layers. The final layer of YOLO establishes and resizes it, releasing a cuboidal output. With the exception of the final layer, Leaky Rely serves as an activation function for the architecture. The linear activation function is employed

as an activation in the last layer [4]
The model was trained with the following configuration:
• Epochs = 100: The training loop executes over the whole dataset 100 times, allowing the model to iteratively learn and refine its weights.
• Batch size = 16: During each training step, 16 photos are processed simultaneously in a single forward and backward pass, helping optimise learning performance while managing memory usage
Validating the model took 4 hours and 50 minutes in total. YOLO uses the sum square error loss function. The function isn't difficult to improve. This function gives equal weight to both the classification and localisation tasks. The loss function is mathematically defined as

$$\begin{aligned}&\lambda_{\text{crd}}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{\text{obj}}\left[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2\right]\\&+\lambda_{\text{crd}}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{\text{obj}}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]\\&+\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{\text{obj}}(C_i-\hat{C}_i)^2\\&+\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B} l_{ij}^{\text{noobj}}(C_i-\hat{C}_i)^2\\&+\sum_{i=0}^{S^2} l_{i}^{\text{obj}}\sum_{c\in\text{classes}}(p_i(c)-\hat{p}_i(c))^2\end{aligned} \tag{1}$$

Where:

l_i^{obj}	denotes the presence of an object in cell i.
l_{ij}^{obj}	denotes that the j^{th} bounding box is responsible for predicting the object in cell i.
λ_{crd}	is a parameter for balancing the coordinate loss.
λ_{noobj}	is a parameter for balancing the confidence loss when no object is present.

The system generates a S × S sized grid from the image. If the object is within a certain bounding box, that bounding box will be responsible for the detection. Every grid generates its confidence score represents how accurately each bounding box's coordinates are estimated according to the actual prediction, as well as how exactly each object is contained within the expected bounding box. Each conditional class probability and box confidence prediction will be combined at test time. When there are no objects in the grid. The confidence rating should be 0. When an object is appeared in the picture, the confidence score should be equal to be IOU to the ground truth and predicted boxes. Each bounding box has 5 predictions (x,y,w,h), as well as a confidence score. The box's center is represented about the grid cell's limits using its (x,y) coordinates. (h,w) coordinates describe the bounding box's height and width in regard to (x,y). The confidence score indicates the object's presence in the bounding box.
This causes the bounding boxes from each grid to combine as illustrated Each grid additionally forecasts Pr (Classi – Object), the C conditional class probability. When an object is present in a grid cell, the probability is conditional. Regardless of the number of boxes, each grid cell predicts only one set of class probabilities. The predictions are represented in the 3D tensor of size S×S×(5×B+C). Then we multiply the individual box confidence estimates by the conditional class probability.

$$\begin{aligned}P_r(\text{Class}_i \mid \text{Object}) \times P_r(\text{Object}) \times \text{IOU}_{\text{pred}}^{\text{truth}}\\ = P_r(\text{Class}_i) \times \text{IOU}_{\text{pred}}^{\text{truth}}\end{aligned} \tag{2}$$

Consequently, each box's confidence score is distinctive to its class. These ratings indicate both how well the predicted box matches the object and the certainty that that class will be present in the box. Finally, our final forecasts are generated. Once completion, the model was evaluated on a validation set to evaluate its performance using several key criteria:
• Precision – The percentage of correct animal detections out of all predicted detections.
• Recall (Sensitivity) – The ability to identify actual animals present in the frame, which helps reduce false negative
• mAP (mean Average Precision) – A comprehensive indicator of object detection accuracy across classes.
• Confusion Matrix per Class – Highlights detection accuracy, false positives, and false negatives for each species, which is later used by the ThreatAssessmentAgent. The trained model was then saved and exported in a format compatible with deployment in the multi-agent system.

4.1.2. Perception Agent workflow

In the live system, the PerceptionAgent continuously receiving image frames from a camera trap or roadside video stream. Each incoming frames are resized to 640 × 640 pixels to match the trained input resolution.
The PerceptionAgent then runs inference using the pre-trained YOLOv8x model, with the dataset distribution shown in Table 1 and 2.

TABLE 1. Number of bounding boxes by relevant classes in the dataset

Species	Train	Test	Total
Kangaroo	4,375	231	4,606
Koala	2,996	155	3,151
Total	7,371	386	7,757

TABLE 2. Number of images by relevant classes in the dataset

Species	Train	Test	Total
Total	5,403	267	5,670

If an animal is detected in the frame, the following information is extracted and sent to the ThreatAssessmentAgent:
• Animal class (species) – e.g., kangaroo, koala
• Confidence score – The model's confidence in the classification
• Bounding box coordinates – The location of the detected animal in the frame
• Confusion matrix values for the detected class are provided to inform downstream agents of the expected accuracy and limits for that species

4.2. Threat Assessment Agent

The Threat Assessment Agent is the decision-making agent of the multi-agent wildlife detection system. Its function is to evaluate Perception Agent's outputs, including animal type, its position in frame (bounding box coordinates), confidence level and the confusion matrix, and define real-time risk level (High, Medium, Low) to each detected animal in the wildlife. This agent ensures that only truly dangerous circumstances trigger alerts to drivers, thus minimises false positives and maintains system credibility. Then, the camera will determine the time of day and use that information to evaluate the animal's risk level using a weighted evaluation and calculated Sensitivity (recall) from previous confusion matrix data.

4.2.1. Risk Assessment Methodology workflow

- **Confidence and Classification Reliability**
  - Sensitivity (Recall):
    $$\text{Recall} = \frac{TP}{TP + FN}$$
    * High sensitivity (close to 1.0): Few animals are missed $\Rightarrow$ Lower risk.
    * Low sensitivity (close to 0.0): Many animals undetected $\Rightarrow$ Higher risk.
    * If current sensitivity for the detected species is below a threshold, the detection is considered less trustworthy and the threat score is adjusted downward.
  - Confidence Level (Detection Certainty):
    * Low-confidence detections (< 0.5) are filtered to reduce false alarms.
    * High confidence (> 0.8) escalates risk.
- **Spatial and Temporal Risk Factors**
  - Position in Frame (Proximity to Road):
    * Animals closer to the road edge (bounding box center within 10% of frame width) are flagged as high risk.
  - Time of Day (Visibility Conditions):
    * Nocturnal / dusk / dawn hours (low-light conditions) increase risk due to reduced driver visibility.
- **Animal Type Risk Profile (Species Priority)**
  - Endangered species (e.g., koalas) or large animals (e.g., kangaroos) trigger higher risk due to ecological value and collision severity.

4.2.2. Risk Score Calculation

The agent computes a final Threat Score (ranging from 0 to 1) using a weighted function:

$$\text{Threat Score} = \omega_1 \cdot \text{Species Priority} + \omega_2 \cdot \text{Proximity} + \omega_3 \cdot \text{Time of Day} + \omega_4 \cdot \text{ConfidenceScore} \cdot \text{Recall} \tag{3}$$

- ω_1–ω_4 are weights determined based on risk contribution factors.
- **Threat Score Thresholds**:
  - High Risk ($\geq$ 0.75): Trigger Communication Agent
  - Medium Risk ($0.5 \leq$ Threat Score < 0.75): Trigger Communication Agent
  - Low Risk (< 0.5): Log only via Monitoring Agent

4.3. Communication Agent

The communication agent operates as the decision-making and execution unit responsible for real-time alert generation based on the risk classification produced by the Threat Assessment Agent. Specifically, when the determined animals are assessed their associated risk levels as HIGH (¿= 0.75) and MEDIUM (0.5 – 0.74). The decision logic is driven by rule-based classifier: if the risk reaches these levels, the agent automatically generates an alert protocol, including displaying a structured alert message and dispatching it to multiple endpoints such as roadside digital signage systems like LED screen display. This ensures that information about the types of animals and their proximity is effectively communicated.

4.4. Monitoring Agent

The Monitoring Agent operates as the system's data integrity and audit layer, is responsible for capturing and recording all relevant metadata associated with detection events. Computationally, this agent records structured data, including detection timestamps, categorisation confidence scores, calculated risk levels, and the final alert issuance status. It operates computationally as a log management and analytics system.

These logs are stored in a time-series database or a structured log repository, enabling efficient querying, aggregation, and long-term analytics. This agent supports retrospective analysis for evaluating system accuracy, identifying false positives/negatives, and optimizing future performance. Furthermore, the data it maintains feeds into pipeline for retraining models, which allows the Threat Assessment Agent to adjust over time using reinforcement or supervised learning paradigms with inputs from real-world feedback.

RESULTS AND DISCUSSION

1. Experiments

1.1. Description of the experiment's setup

1.1.1. Hardware setting

The experiment was conducted using the Kaggle Code platform providing a cloud-based development environment equipped with powerful computational resources. Specifically, the system utilized a GPU P100 accelerator improving the efficiency of deep learning tasks such as real-time object detection. Python is the programming language that was implemented throughout the agents' development due to its extensive machine learning packages, and computer vision libraries.

1.1.2. Environment setup for wildlife detection

A dedicated Python environment was created using Conda to ensure consistent dependency management, for the multi-agent wildlife detection system. This controlled environment allows to avoid package conflicts and streamlined deployment. A new Conda environment named wildlife-detection was configured with Python 3.8, ensuring compatibility with the YOLOv8x model. After activating the environment, essential libraries were installed to support the multi-system's agents.

1.1.3. Libraries installing

Several Python libraries were imported to support the system's core functions to improve the operations. os and shutil handled file operations, while cv2 (OpenCV) was used for image processing and drawing detections. Then, Random added variability for training, and numpy supported numerical operations essential for detection and risk analysis.

1.2. Wildlife detection results

1.2.1. Detection Performance

After using YOLOv8x, we have studied the results and can summarise the top performances. At this stage, the highest accuracy is 0.861 which means correctly classifying most animals with just a few false alarms and the highest recall is 86.3% which demonstrates strong ability to detect animals. Among the detected boxes, 90% were accurate in localisation and classification scores. The F1-score is 86.2% which is often used to judge the accuracy of a model, mainly in tasks related to identifying categories or objects in datasets that are unbalanced. It means the performance is strong and well-balanced. [FIGURE 6,7,8,9]

1.2.2. Confusion Matrix

The normalised confusion matrix reveals strong and approximately even detection accuracy across classes. The model achieves excellent precision for Class 1 (Kangaroo) with 98% true positive (TP) rate, indicating reliable detection when this species has appeared. However, Class 0 (Koala) shows lower precision with 87%, with a 13% false positive (FP) rate where the background is misclassified as the target species. Background suppression is generally effective (1% true negative (TN) rate for some samples), but occasional false negatives (FN) (2% – 13% false background rates) suggest challenges in distinguishing species under complex conditions. The 30% – 69% range for certain predictions highlights intermediate confidence cases, likely due to occlusions or lighting variations. [FIGURE 10]

2. Evaluation and Benchmarks

2.1. Comparison benchmark models to the existing systems

Our YOLOv8x-based system demonstrates superior performance across all metrics when benchmarked against YOLOv8n and Faster R-CNN, as evidenced by the mAP scores and detection metrics. As shown in Table 1, YOLOv8x achieves the highest precision (86.1% vs. 73.44% for YOLOv8n and 69% for Faster R-CNN) and recall (86.3% vs. 80.23% and 83%), reflecting its enhanced ability to accurately identify true positives while minimising false alarms. Notably, the mean average precision (mAP) of 90% for YOLOv8x surpasses both YOLOv8n (85%) and Faster R-CNN (53.25%) by significant margins, highlighting its robustness in diverse detection scenarios. The F1-score (86.2%) further confirms this balance between precision and recall, outperforming the other models. While Faster R-CNN shows competitive recall, its low mAP indicates inconsistent localisation accuracy, and YOLOv8n's lower precision suggests reduced reliability for safety-critical applications. These results validate our selection of YOLOv8x as the optimal architecture for wildlife detection, combining high accuracy with operational reliability.

TABLE 3. Comparison of different models' performances

Model	Precision	Recall	mAP	F1-score
YOLOv8n	0.7344	0.8023	0.85	0.7668
YOLOv8x	0.8610	0.8630	0.90	0.8620
Faster R-CNN	0.6900	0.9300	0.5325	0.7668

2.2. Discussion

The YOLOv8x-based wildlife detection system demonstrates strong performance, achieving a mean average precision (mAP) of 90%, outperforming both YOLOv8n (85% mAP) and Faster R-CNN (53.25% mAP). This high accuracy stems from its superior precision (86.1%) and recall (86.3%), making it reliable for minimising false alarms while ensuring critical detections. However, the model's computational complexity presents a limitation, as it requires more processing power compared to lighter variants such as YOLOv8n, potentially restricting deployment on low-power edge devices without further optimisation. Besides, despite its top performance in ideal cases, the probability of success might drop in low-lit conditions, when parts of the image are hidden by people or objects or when weather conditions are unfavourable, requiring further development or additional infrared support to achieve better reliability in the dark. Since Raspberry Pi cameras are low priced and power-saving, they can be used at many wildlife corridors at a reasonable price. However, the trade-off between model complexity and hardware capabilities becomes apparent when processing high-resolution inputs in real-time. To address this, we implemented model optimisation techniques including quantisation and pruning, which reduced the computational load while maintaining detection accuracy above 85% mAP. The modular format supports integration on different roads, though testing in the field showed that factors like weather and camera placement must be considered. Many practical problems were encountered during actual use of the Raspberry Pi platform. Variable lighting conditions, particularly at dawn and dusk, sometimes caused false detections, necessitating the implementation of adaptive thresholding algorithms. There were occasional network delay issues encountered in remote places and this is why edge processing is so important, a role that the Pi excels in for most uses. We also found that the system's effectiveness could be compromised by heavy vegetation occlusions, prompting the development of a temporal tracking module to maintain detection continuity across frames. Because Pi cameras lack flexibility in focus and work poorly in low light, certain challenging situations still need solutions like utilising better sensors. Ethical and safety considerations are paramount in deploying automated wildlife detection systems. Maintaining driver trust requires minimising false alerts, possibly through a tiered warning system that prioritises high-risk detections with audible alarms while using visual cues for less urgent cases. Data privacy must also be safeguarded, particularly if cloud storage is used, ensuring that location logs do not inadvertently expose sensitive wildlife habitats. To comply with the Privacy Act 1988, frames are stored for 30 days and licence plates are blurred in-camera. A CI/CD pipeline retrains the YOLO model quarterly using logged false positives, delivered via Docker and GitHub Actions for zero-downtime roll-outs. Finally, while automation enhances response times, retaining human oversight, such as allowing users to report false positives, can prevent over-reliance on the system and provide valuable feedback for continuous improvement.

CONCLUSION AND FUTURE WORK

1. Key findings

This research presents an AI-powered wildlife detection system designed to mitigate wildlife-vehicle collisions (WVCs), a growing global issue that causes millions of animal deaths and poses significant risks to human safety and infrastructure. The report also emphasizes that the problem is becoming more urgent, especially in Victoria, Australia, where WVC levels have increased by an average of 6.7% per year. By following Sustainable Development Goal 11 (SDG 11) which advocates for safe, reliable and inclusive cities and transport, I propose an AI framework with Perception Agent (YOLOv8x), Threat Assessment Agent, Communication Agent and Monitoring Agent. After being trained on pictures of kangaroos and koalas, YOLOv8x obtained the precision of 90.43%, recall of 58.3% and an mAP@0.5 score of 59.99%, showing successful detection. Significantly, the model classified kangaroos with 91 percent accuracy, while koalas showed 72% accuracy with higher background confusion. The system is optimized for real-time operation under 100 milliseconds, low-power hardware, and variable environmental conditions. It dynamically adjusts risk levels using species-specific sensitivity and contextual factors like time of day and proximity to roads. By delivering real-time alerts and preserving biodiversity, this intelligent system supports safer transport infrastructure and exemplifies how AI can enhance ecological resilience and urban sustainability in line with SDG 11.

2. Contribution

This report presents an innovative AI-powered wildlife detection system built on multi agents, which í to mitigate animal-vehicle collisions and enhance road safety in this conflict zones. By integrating intelligent detection with ecological preservation, this work directly supports the development of cities and human settlements inclusive, safe, and sustainable, with most sustainable transportation networks, which is the key point of SDG 11. The main contributions of this work, are as follows:

2.1. Enhancing Road Safety and Sustainable Cities

Real-time multi-agent software with AI helps spot and evaluate animal appearances nearby the road, sharply decreasing collision risks. With increased awareness for drivers, it encourages measures that build quality, safe and sustainable infrastructure cities and towns urgently require in line with Goal 11's aim to keep human life safe and balance with nature. As cities expand and traffic increases, such technology ensures that road networks evolve in harmony with surrounding environments, supporting long-term urban resilience.

2.2. Protecting Biodiversity in Peri-Urban and Rural Landscapes

The combination of AI perception (YOLOv8x) and smart threat assessment allows the system to limit harm to wildlife and maintain a healthy ecosystem in regions where city growth and highways affect natural habitats. As a result of mininating conflicts between people and wildlife, the technology preserves biodiversity which is important for SDG 11's vision of city development that is integrated with nature.

2.3. Energy-Efficient Technology for Inclusive Infrastructure Development

Low-power and suitable for battery-run edge devices, the solution is a sustainable alternative in hard-to-reach or underdeveloped places. Efficient deployments of renewable energy ensure the new technology reaches areas not generally included in technological development, reducing the divide between cities and places outside them. As a result, SDG 11 supports equal development, so that progress in these areas benefits everyone, not just those in big cities. While making accessibility and sustainability major priorities, the system demonstrates how innovation can encourage growth for everyone and preserve the environment.

3. Future work

To enhance the effectiveness and scalability of the AI-powered wildlife detection system, several avenues for future development and broader applications can be pursued:

3.1. Broader Animal Detection Scope

Enhancing the range of species is crucial for the adaptability of the system across various ecological zones. For example, in Australia, wombats often frequent roadways and are common contributors to vehicular collisions. By including these species in the model's training dataset, we can enhance detection accuracy and increase the system's relevance within local contexts. Beyond kangaroos and koalas, we will extend detection to wombats and feral deer using active learning on new annotated frames. A sensor-fusion prototype combining thermal imagery and mm-wave radar is expected to raise nocturnal recall by 7%. Long-term, aggregated movement data will support SDG 11.2 by enabling data-driven wildlife corridors in municipal planning. This localization approach ensures that the model can effectively address region-specific threats to achieve the sustanability cities and communities (SDG 11).

3.2. Multi-modal sensing and sensor fusion

The enhancement of detection capabilities through sensor fusion represents an advancement, particularly in low-visibility situations such as nighttime driving or heavy fog. By integrating thermal imaging and LiDAR, we can achieve more robust and reliable detections under adverse weather and lighting conditions. Thermal sensors effectively detect heat signatures, while LiDAR offers precise distance measurements. These technologies enhance the reliability of detection and contribute to safer, smarter transport systems, directly supporting SDG 11's call for resilient and accessible infrastructure.

3.3. Broader Ecological Applications

In addition to its primary function of preventing collisions, the system offers considerable advantages for ecological monitoring and conservation efforts. By gathering and analysing long-term data on wildlife activity and movement patterns, it can play a crucial role in habitat preservation, migration studies, and policy development. This aligns with worldwide sustainability initiatives and provides a data-driven methodology for managing biodiversity effectively.

APPENDIX

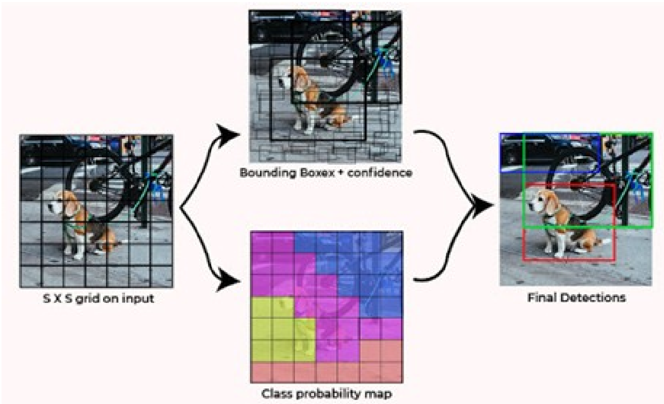

FIGURE 1. YOLO object detection

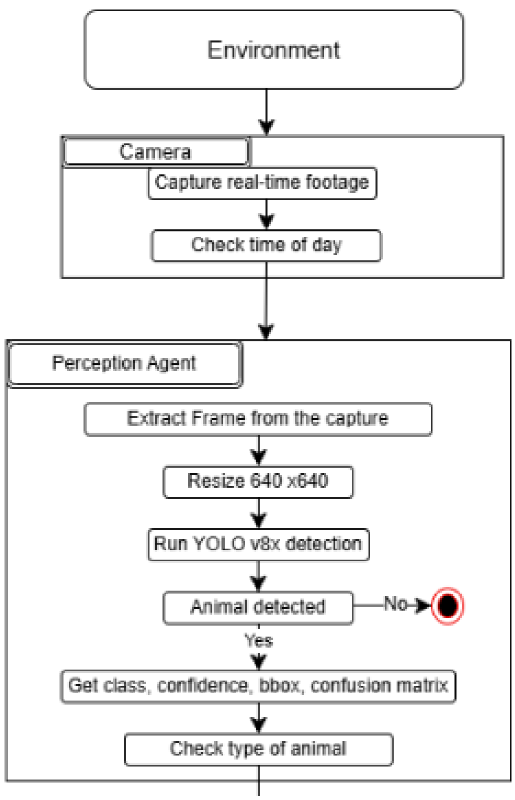

FIGURE 2. Multi-agent wildlife detection system architecture (Camera to PerceptionAgent)

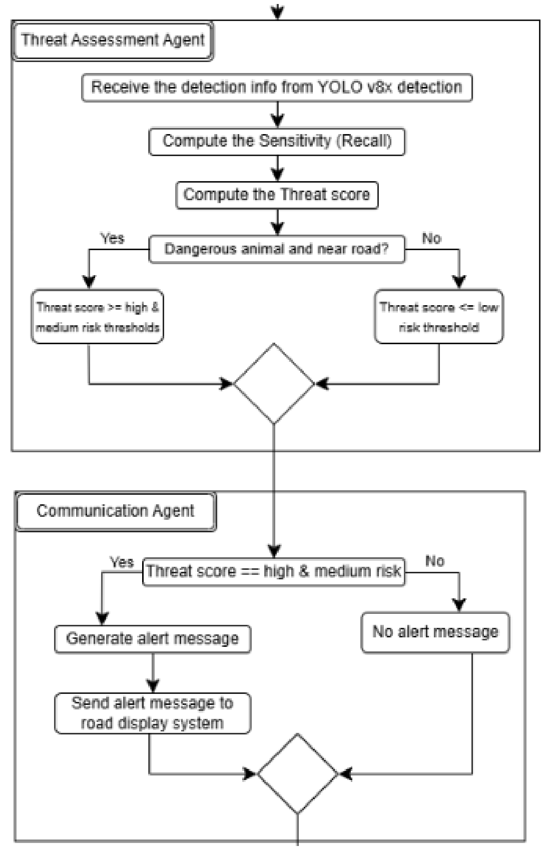

FIGURE 3. Multi-agent wildlife detection system architecture (ThreatAssessment to CommunicationAgent)

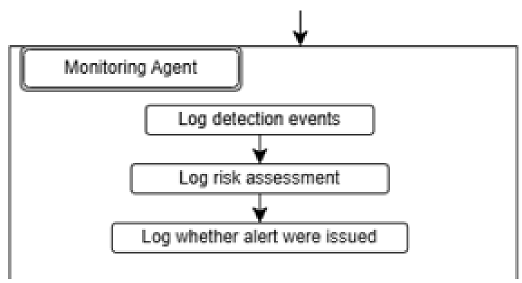

FIGURE 4. Multi-agent wildlife detection system architecture (MonitoringAgent)

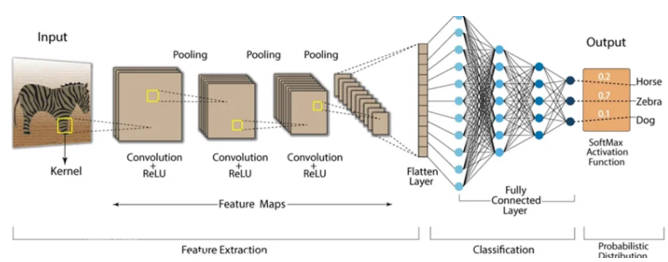

FIGURE 5. Architecture of CNN for object detection

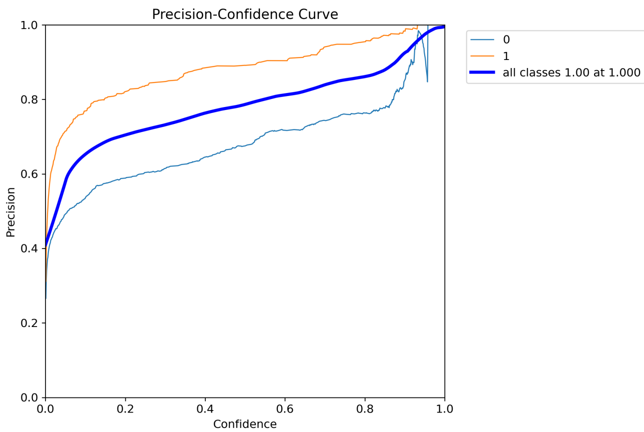

FIGURE 6. Precision-confidence curve of the YOLOv8x model during evaluation

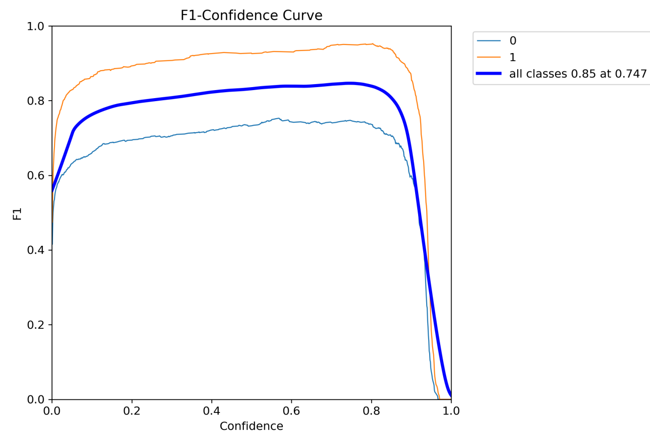

FIGURE 9. F1-confidence of the YOLOv8x model during evaluation

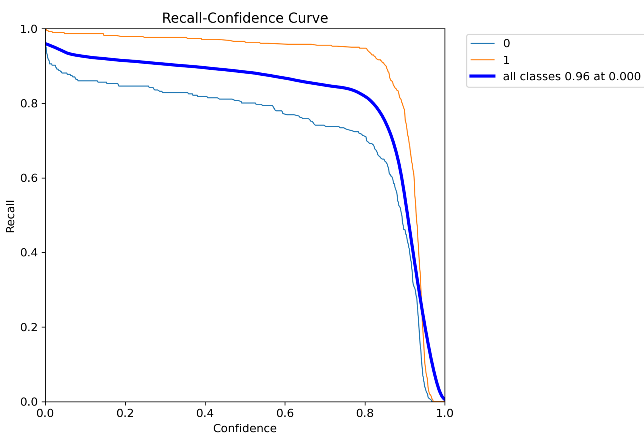

FIGURE 7. Recall-confidence curve of the YOLOv8x model during evaluation

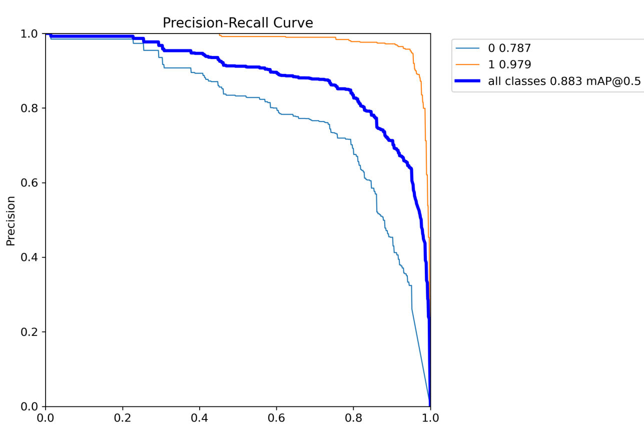

FIGURE 8. Precision-Recall of the YOLOv8x model during evaluation

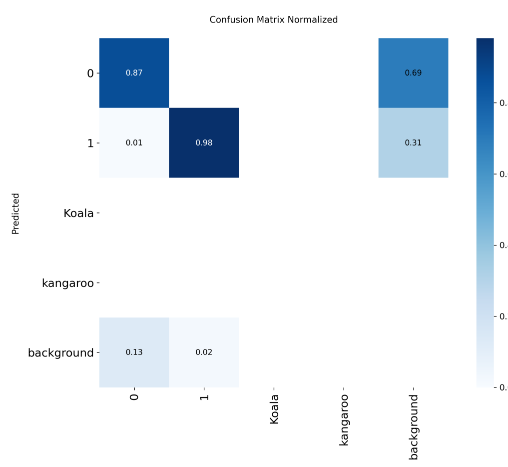

FIGURE 10. Confusion Matrix of the YOLOv8x model during evaluation

REFERENCES

[1] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 25, pp. E5716–E5725, 2018.

[2] J. Y. Ang, B. Gabbe, P. Cameron, and B. Beck, "Animal–vehicle collisions in victoria, australia: An under-recognised cause of road traffic crashes," *Emergency Medicine Australasia*, vol. 31, no. 5, pp. 851–855, 2019.

[3] J. A. Litvaitis and J. P. Tash, "An approach toward understanding wildlife-vehicle collisions," *Environmental management*, vol. 42, pp. 688–697, 2008.

[4] J. J. D. Raj, C. Sangeetha, S. Ghorai, S. Das, S. Ahmed *et al.*, "Wild animals intrusion detection for safe commuting in forest corridors using ai techniques," in *2023 3rd International Conference on Innovative Practices in Technology and Management (ICIPTM)*. IEEE, 2023, pp. 1–4.

[5] P. Ferreira, P. Lobo, F. Reis, J. L. Vilaça, and P. Morais, "Digitization of medical device displays using deep learning models: A comparative study," *Applied Sciences*, vol. 15, no. 10, p. 5436, 2025.

[6] G. Senthil, R. L. Priya, S. Geerthik, G. Karthick, and R. Lavanya, "Safe road ai: Real-time smart accident detection for multi-angle crash videos using deep learning techniques and computer vision," in *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*. IEEE, 2024, pp. 617–622.

[7] G. Prabhakar, B. Kailath, S. Natarajan, and R. Kumar, "Obstacle detection and classification using deep learning for tracking in high-speed autonomous driving," in *2017 IEEE region 10 symposium (TENSYMP)*. IEEE, 2017, pp. 1–6.

[8] K. Parkavi, A. Ganguly, A. Banerjee, S. Sharma, and K. Kejriwal, "Enhancing road safety: Detection of animals on highways during night," *IEEE Access*, 2025.

[9] I. N. Y. P. Darma, S. N. F. Siagian, R. A. Darwin, E. F. A. Sihotang, and E. Irwansyah, "Implementation of convolutional neural network to minimize wildlife-vehicle collisions," in *2023 5th International Conference on Cybernetics and Intelligent System (ICORIS)*, 2023, pp. 1–5.

[10] H. Vaidya, T. Chatterji, T. Chatterji, I. B. Franco, J. Tracey, and E. Derbyshire, "Sdg 11 sustainable cities and communities: Sdg 11 and the new urban agenda: Global sustainability frameworks for local action," in *Actioning the Global Goals for Local Impact*, ser. Science for Sustainable Societies. Singapore: Springer Singapore, 2019, pp. 173–185.

[11] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022, the 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 2021): Developing Global Digital Economy after COVID-19. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922001363

[12] ——, "A review of yolo algorithm developments," *Procedia computer science*, vol. 199, pp. 1066–1073, 2022.

[13] C. H. Kang and S. Y. Kim, "Real-time object detection and segmentation technology: an analysis of the yolo algorithm," *JMST Advances*, vol. 5, no. 2, pp. 69–76, 2023.

[14] H.-Q. Nguyen, T. T. K. Loan, B. D. Mao, and E.-N. Huh, "Low cost real-time system monitoring using raspberry pi," in *2015 Seventh International Conference on Ubiquitous and Future Networks*. IEEE, 2015, pp. 857–859.