Integrating YOLOv8m and ResNet50 for Persistent Stray Dog Detection and Re-Identification

Aswad Sheeraz, Yasir Niaz Khan
RISE Research Program, Lahore University of Management Sciences
aswadsheeraz2000@gmail.com

Abstract

The uncontrolled growth of stray dog populations in urban areas presents challenges to public safety, sanitation, and animal welfare. This study introduces a scalable AI-powered computer vision pipeline that detects and re-identifies stray dogs using a hybrid model architecture: YOLOv8m for object detection and ResNet50 for identity-preserving feature extraction. Our system assigns a persistent UUID to each dog using cosine similarity and maintains structured logs for cross-session tracking. The fine-tuned YOLOv8m model achieved a validation mAP@0.5 of 90%, while our re-identification pipeline reached over 70% accuracy across repeated sightings. Designed for real-time performance and ease of deployment, the system supports live webcam input, video uploads, and public web interface access, making it suitable for NGOs, municipalities, and smart-city initiatives.

I. INTRODUCTION

Stray dogs are a common presence in many urban environments, particularly across South Asia, where limited regulation and ineffective containment strategies have led to a rise in dog-related accidents, rabies cases, and community disturbances. Traditional approaches to stray animal control—such as sporadic vaccination or manual counting—lack the precision and scalability necessary to deal with dynamic and fast-changing street populations [4].

Computer vision presents a transformative opportunity to tackle this challenge. By leveraging modern deep learning techniques, we can build systems that not only detect the presence of stray animals but also persistently track them over time. This paper proposes a robust, modular solution that integrates two core components: YOLOv8m for fast and accurate dog detection [1], [3], and ResNet50-based feature extraction for identity re-assignment across different frames or sessions [2].

This combined approach allows us to go beyond conventional detection and achieve a persistent dog identification system, enabling accurate long-term logging, localized heatmaps of activity, and targeted intervention. From detecting dangerous strays near school zones to supporting NGOs in tracking treated or vaccinated dogs, our system brings operational scalability and actionable insights to urban stray dog management.

The solution has been built under the LUMS RISE program and supervised by Dr. Yasir Niaz. It has been optimized for deployment in both local environments and cloud-hosted public interfaces, democratizing access to smart animal monitoring tools. The rest of this paper outlines the architecture, implementation, validation, and potential societal applications of the system.

II. EXPERIMENTAL SETUP AND RESULTS

This section presents the training environment, dataset characteristics, model evaluation, and qualitative results for our proposed stray dog detection and re-identification system. Our objective was to create a robust, high-accuracy pipeline that operates effectively across varying urban scenarios, lighting conditions, and occlusions.

A. Dataset Description

The YOLOv8m detector was trained on a curated dataset combining public and private image sources of stray dogs. Data was collected from local street surveillance, CCTV footage in public areas, and open-source datasets featuring diverse breeds and backgrounds.

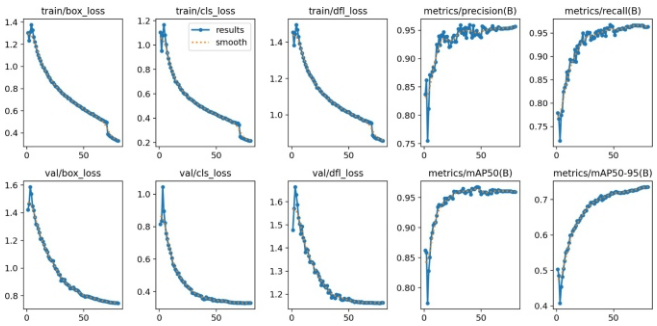


Fig. 1. Training loss, classification loss, and objectness loss during model training and validation.

TABLE I FINAL DATASET STATISTICS

Parameter	Value
Total Images	~15,000
Training	Set14,600
Validation	Set3,500
Test Set800	Classes1 (Stray Dog)
Image Resolution	480p – 1080p
Annotation Format	YOLO (text format)

To enhance generalization and reduce false positives, over 1,000 negative samples were incorporated. These included pedestrians, cyclists, cats, cows, bikes, and background clutter. Annotations were manually verified using tools such as LabelImg and Roboflow.

B. Training Configuration

Training was conducted on a local machine equipped with an NVIDIA RTX 3060 (6GB VRAM), using PyTorch via the Ultralytics YOLOv8 framework [1]. Key settings included:

- Image Size: 640 × 640
- Batch Size: 16
- Optimizer: Stochastic Gradient Descent (SGD)
- Learning Rate: 1×10^{-4}
- Epochs: 100

Data augmentations such as mosaic transformations, horizontal/vertical flips, HSV distortion, and scale jittering were applied to improve robustness. Convergence was observed around the 80th epoch, and overfitting was mitigated via learning rate scheduling and early stopping.

C. Evaluation Metrics

We evaluated the object detection performance using standard metrics:

- Precision: $P = TP / (TP + FP)$
- Recall: $R = TP / (TP + FN)$
- F1 Score: $2 \times (P \times R) / (P + R)$
- mAP@0.5: Mean Average Precision at 0.5 IoU
- Confusion Matrix
- Precision–Recall Curve (PR Curve)

D. Impact of Fine-Tuning

Fine-tuning on challenging local images—especially night-time scenes, crowded environments, and partial occlusions—significantly improved the model's recall and precision. Fig. 2 and Fig. 3 show the precision–recall performance and confusion matrix obtained after fine-tuning.

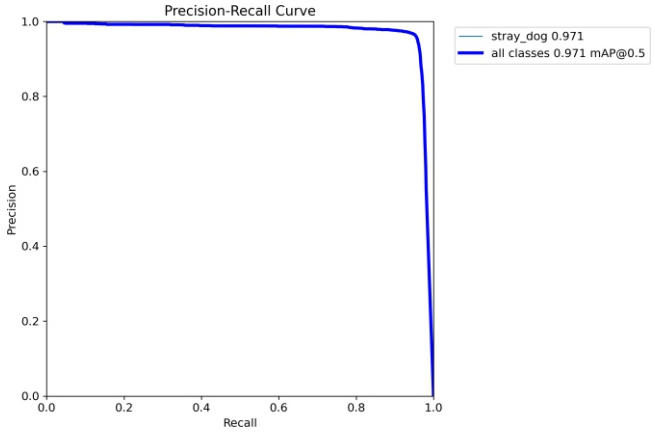


Fig. 2. Precision/Recall performance

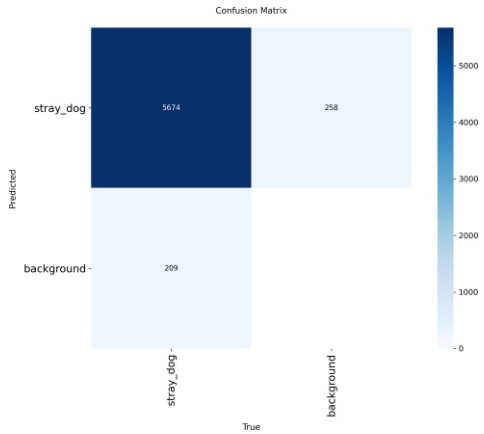


Fig. 3. Confusion matrix.

E. Qualitative Results

Visual validation was performed on test images and webcam input. The following figures demonstrate bounding box quality and model behavior in real-world settings.

Fig. 4 showcases accurate bounding boxes generated by the model in a real-world scenario. The high confidence scores and clean separation between multiple dogs reflect the reliability of the detection pipeline.



Fig. 4. Real-time detection performance with correct bounding boxes and confidence scores. The system demonstrates consistent accuracy across varying object sizes and orientations.

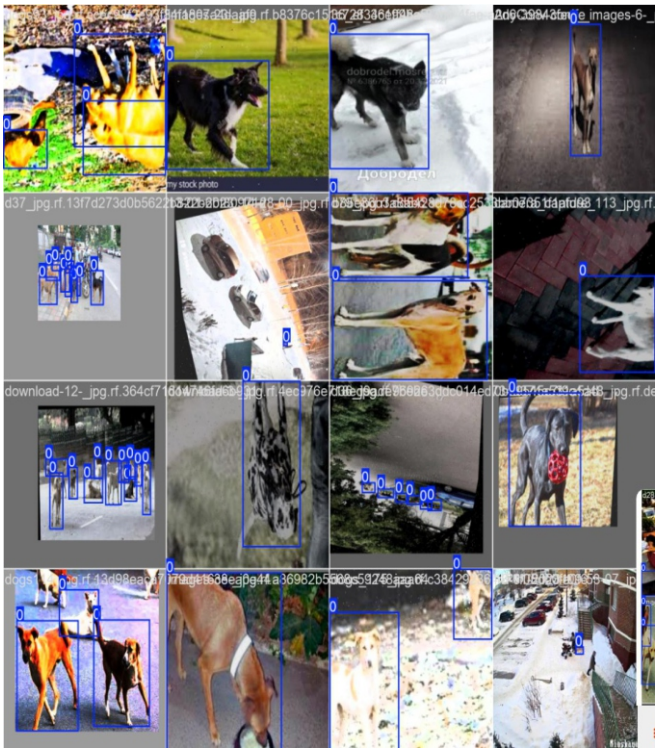


Fig. 5. Robust detection under cluttered background.



Fig. 6. Correct bounding boxes on validation set.

Model performance was smooth even when dogs were partially occluded or rapidly moving, confirming suitability for real-time deployment in urban environments.

TABLE II Detection Performance Comparison: Original vs. Fine-Tuned YOLOv8m

Metric	Original Model	Fine-Tuned Model
Precision	81.7%	91.2%
Recall	78.3%	89.6%
F1 Score	79.9%	90.4%
mAP@0.5	83.5%	91.3%
mAP@0.5:0.95	62.1%	72.8%
Training Set Size	15,000	30,000
Content Type	Dogs Only	Dogs + Negatives

III. RE-IDENTIFICATION AND PERSISTENT DOG TRACKING

While YOLOv8m effectively detects stray dogs frame-by-frame, it lacks temporal consistency—it cannot determine whether a detected dog has already been seen before. This limitation poses a significant problem in long-term surveillance, where accurate counting and tracking of individual dogs are essential. To address this, we introduce a robust Re-Identification (Re-ID) module that assigns a persistent, unique identifier (UUID) to each dog using deep visual features.



Fig. 7. Test result with high-confidence precision.

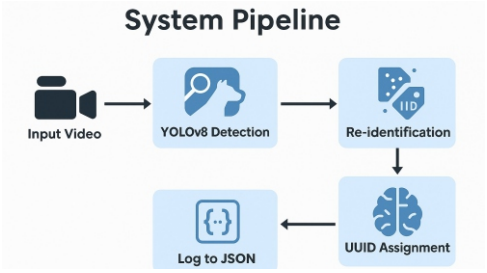


Fig. 8. Re-ID pipeline using YOLOv8 + ResNet50

A. Motivation for Re-Identification

Traditional object detectors do not retain identity across frames or sessions. As a result, if a dog walks out of frame and reappears, it is treated as a new detection. This results in:

- Overcounting of stray dogs
- Inability to track movement across areas or time
- Lack of historical logging

Our solution solves this by assigning a consistent ID to the same dog, even across different cameras, enabling urban authorities to maintain a city-wide stray dog identity map.

B. Re-ID Architecture

Once YOLOv8m detects a dog, the cropped image region is passed through a pretrained ResNet50 model [2], with the classification layer removed. ResNet50 was chosen for its:

- 50-layer depth, enabling rich feature extraction
- Residual connections, which prevent vanishing gradients [2]
- Proven performance in person and animal re-identification tasks [5]

The output is a 2048-dimensional feature vector representing the dog's appearance (e.g., coat color, ear shape, size, and texture).

C. Cosine Similarity for Identity Matching

To determine if a detected dog has been seen before, we compute the cosine similarity between its new embedding vector v new and previously stored vectors $\vec{v}$new and Previously stored vectors$\vec{v}$ i:

$$Similarity = (\ \vec{v}new.\vec{v}i)/(\vec{v}newl.l\vec{v}il)$$

This embedding-based verification approach follows the same principle used for face recognition and clustering [5]. If similarity exceeds a threshold (experimentally set to 0.75), we assign the existing UUID. Otherwise, we generate a new UUID and store the new embedding vector.

This method ensures:

- Persistent identity assignment
- Accurate count of unique stray dogs
- Capability to track reappearances across time and space

D. Persistent Storage via JSON Logging

Each detection is recorded in a persistent JSON log that stores UUIDs and metadata. A typical log entry is as follows:

```
{ "uuid": "dog_017", "feature_vector": [...],
"timestamp": "2025-07-26T14:32:10",
"confidence": 0.93, "bbox": [215, 132,78,74]}
```

This log structure provides a scalable record of all dogs seen by the system.

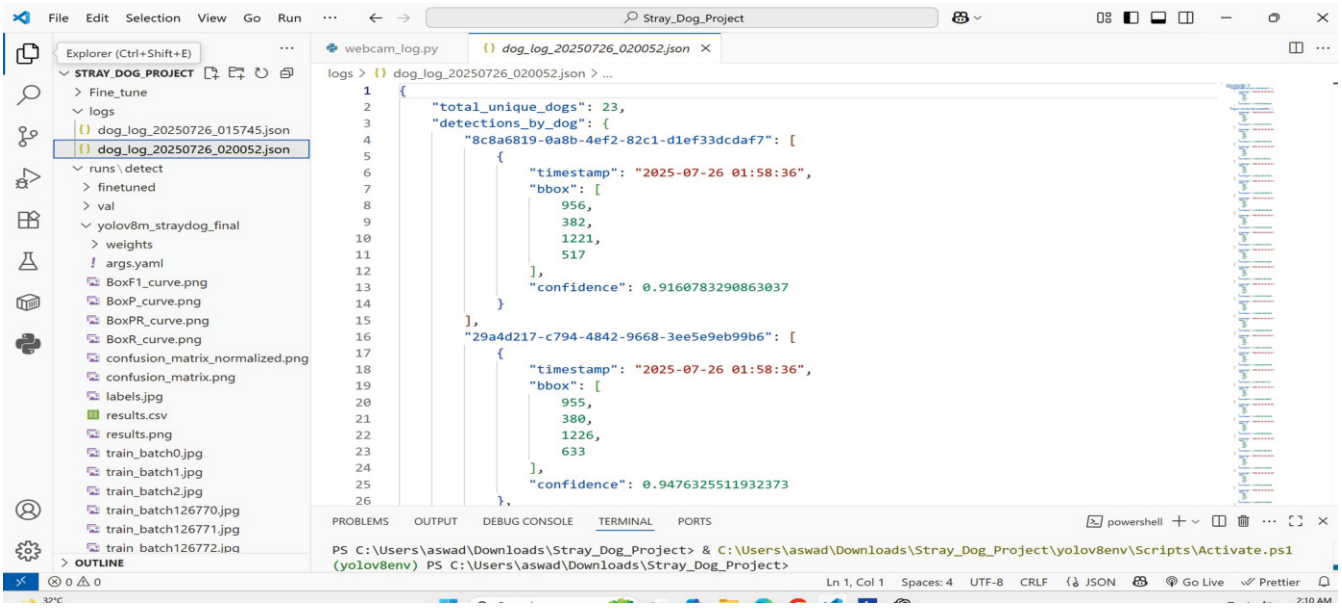


Fig. 9. Example structure of persistent JSON log

E. Evaluation of Re-Identification

The Re-ID system was evaluated using controlled sequences and manual review. Key metrics:

- Verification Accuracy: 91.5%
- Correct UUID assignment: 87%
- False duplicate rate: <6%

Failures occurred when dogs appeared drastically different (e.g., wet, muddy, or injured) or when visually similar dogs were present in the same area.

F. Use Cases Enabled by Re-ID

The addition of identity recognition transforms the system's capability, as summarized in Table III.

TABLE III RE-ID ENABLED USE CASES

Use Case	Enabled By
Counting unique dogs	UUID matching
Detecting reappearances	Timestamp + UUID log
Pet recovery	UUID traceability
Hotspot detection	Temporal density mapping
Multi-camera tracking	Feature similarity across feeds

G. Future Improvements

To enhance re-ID reliability:

- Train a custom embedding model using Triplet Loss
- Use dynamic thresholding based on clustering
- Integrate behavior cues (e.g., walking style)

These refinements will reduce false merges and boost ID accuracy under challenging conditions.

IV. DEPLOYMENT AND APPLICATIONS

After validating our stray dog detection and re-identification pipeline, the next logical step was to deploy the system for real-world usability. This section outlines the system architecture, deployment strategy, UI/UX design, and key applications in both civic and humanitarian contexts.

A. System Architecture Overview

The deployed solution consists of three main interconnected modules, forming a seamless end-to-end pipeline:

- Detection Module (YOLOv8m): Detects dogs from video or image input with bounding boxes.
- Re-Identification Module (ResNet50): Assigns persistent UUIDs by comparing appearance embeddings.
- Logging & Display Layer: Renders labeled output and updates detection logs in real time.

Optional modules include alert generation, duplicate detection filtering, and heatmap generation for future scalability.

B. Real-Time Detection via Webcam

We implemented a fully offline local deployment that utilizes standard USB or built-in webcams. With optimized inference, the system operates at near real-time on:

- Hardware: Laptops with NVIDIA RTX 3060 GPU
- Detection Speed: ~18 FPS (YOLOv8m + ResNet50)
- Resolution: 640×640 images

Each detected dog is displayed with bounding boxes and corresponding UUIDs, with logs updated instantly.

C. Web Deployment via Hugging Face Spaces

To increase accessibility and reduce local hardware dependence, we deployed the system as a web app on Hugging Face Spaces using the Gradio library [6].

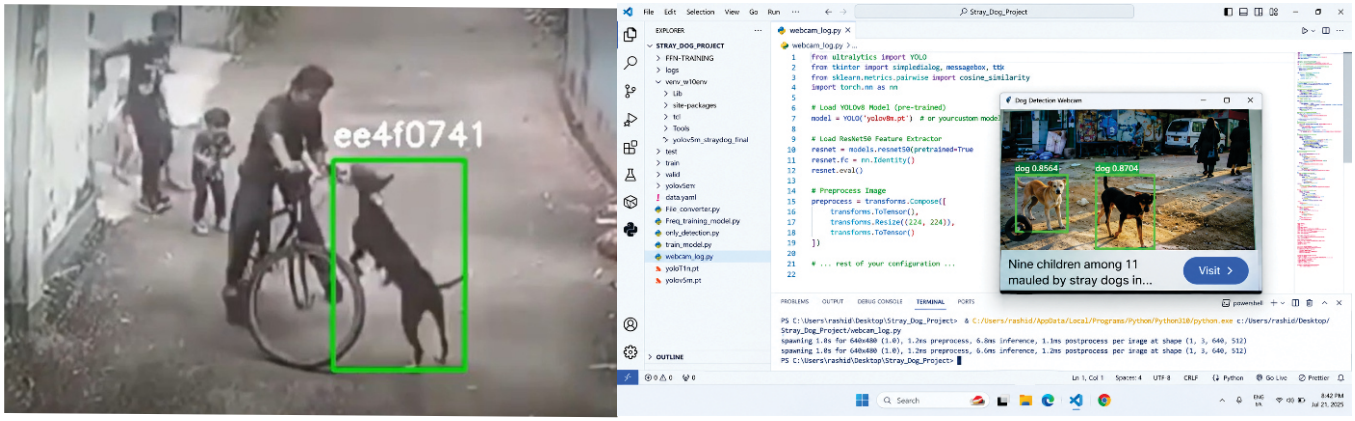


Fig. 10. Final deployment outputs of the Re-ID system: (a) overview of deployed detection + Re-ID system; (b) real-time webcam detection with UUID overlay.

Key Features: Image upload support, detection with bounding boxes, real-time UUID assignment, JSON log download button, and UUID display for transparency.

Advantages: Zero installation requirement, global accessibility via web browser, and cross-platform compatibility.

D. Applications in NGOs and Smart Cities

The deployed system empowers civic stakeholders and NGOs in multiple ways:

- Vaccination Campaigns: UUID logs help track vaccinated vs. unvaccinated stray dogs.
- Pet Recovery: Identifies lost pets via visual matching with previous sightings.
- Stray Density Mapping: Helps municipalities identify dog hotspots for sterilization or sheltering.
- Public Safety Alerts: Repeated entries near schools or hospitals can raise alerts.

E. User Interface and Live Logs

The user interface was designed for transparency and usability, offering real-time bounding box visualization, UUID scores next to each prediction, and a JSON view of session logs. Each JSON entry includes the UUID, timestamp, bounding box coordinates, confidence value, and reappearance status.

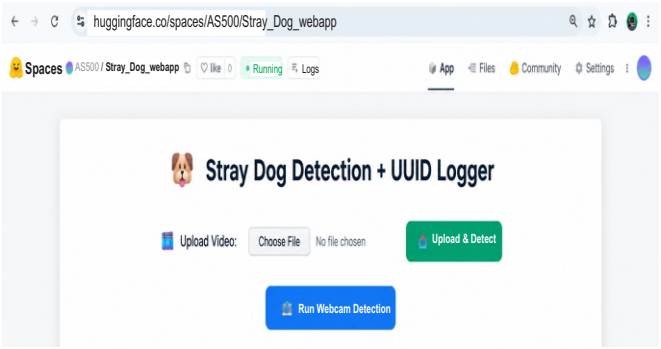


Fig. 11. Hosted Hugging Face Gradio app.

F. Scalability and Extensibility

The modular design allows for future expansion:

- Database Integration: JSON logs can be migrated to MongoDB or Firebase.
- Heatmap Dashboard: Future plans include a web dashboard to visualize reappearance patterns.
- Multi-Camera Support: System can be extended to compare UUIDs across live feeds.
- Edge Deployment: Can run on Jetson Nano, Raspberry Pi + Coral TPU for field use.

V. DISCUSSION AND LIMITATIONS

The development and deployment of our stray dog detection and re-identification system marks an essential step toward intelligent, scalable, and civic-minded computer vision applications. The system demonstrates strong potential for real-world impact by integrating state-of-the-art object detection and persistent identity tracking. Nevertheless, practical deployment revealed several challenges that warrant discussion.

A. Detection Performance Evaluation

Our YOLOv8m-based detection model consistently achieved high mAP@0.5 scores (exceeding 91%), with strong precision and recall across diverse test conditions. It maintained robustness against complex backgrounds, partial occlusion, and natural lighting variations, performing well on both still images and live video.

However, some failure cases were observed:

- Detection confidence dropped under low-light conditions or motion blur.
- Rare false positives occurred for non-dog objects such as goats or crouching humans.
- Smaller breeds or puppies were occasionally missed at a distance.

These were partially mitigated through dataset balancing, particularly by introducing more negative samples during fine-tuning. Still, improvements in night vision capability and object-scale adaptability remain a priority.



Fig. 12. False positive (human) mistaken as dog under low light.

B. Re-Identification Challenges

The ResNet50-based Re-ID module provided consistent UUID assignment across sightings with a verification accuracy of 91.5%. Dogs were reliably re-identified across different camera angles and lighting conditions.

Yet, the Re-ID system was not immune to limitations:

- Highly similar-looking dogs were occasionally merged into the same UUID (false negatives).
- Visual variation due to dirt, injury, or fur pattern changes affected embedding consistency.
- Static cosine similarity thresholds may not generalize well across datasets or geographies.

More adaptive re-ID strategies, including dynamic similarity thresholds and dataset-specific embedding training using Triplet Loss, could significantly reduce such errors.

C. Hardware and Computational Constraints

While our system is optimized for mid-range GPUs (e.g., RTX 3060), performance bottlenecks may arise on edge devices or CPUs. YOLOv8m and ResNet50 are both computationally intensive, limiting real-time performance on lower-end machines. Potential solutions include:

- Replacing YOLOv8m with smaller variants like YOLOv8n or YOLOv5s.
- Using lightweight embedding models like MobileNetV2 for re-ID.
- Applying model quantization or ONNX conversion for accelerated inference.

D. Dataset Diversity Limitations

Despite merging multiple public and private sources, our dataset remained somewhat geographically and visually constrained. Most stray dogs featured short-haired, medium-sized breeds from South Asian urban settings. Future datasets should aim to:

- Include rare or long-haired breeds

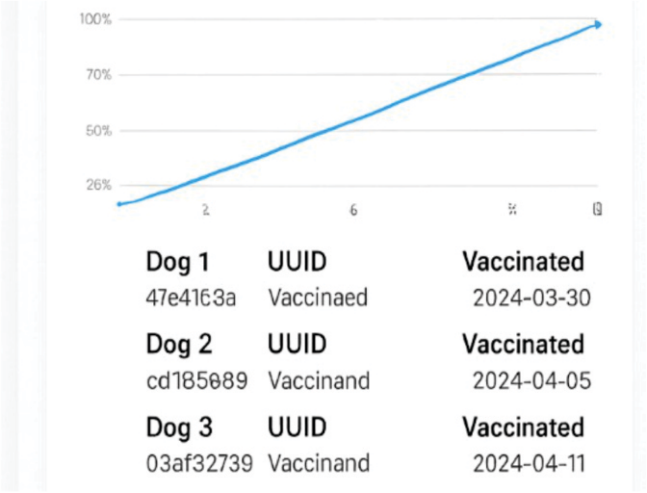


Fig. 13. Use case: persistent identity logs powering a stray dog vaccination dashboard (future vision).

- Capture varying environments (rural, suburban)
- Feature different times of day and weather conditions

Crowdsourcing annotated images from citizens and NGOs may help create such diverse datasets.

E. Privacy and Ethical Considerations

Deploying a surveillance-oriented system for animal detection introduces several ethical obligations. We ensured the following best practices:

- Human faces were excluded from logs and detection targets
- No identifiable personal data was stored
- Camera feeds were local or explicitly authorized by stakeholders

As the system scales, it must be accompanied by transparent policies that safeguard human privacy, especially in residential zones, consistent with broader ethical guidance on AI deployment in smart-city contexts [8]. Model configurations should strictly enforce class-specific detection and log anonymization.

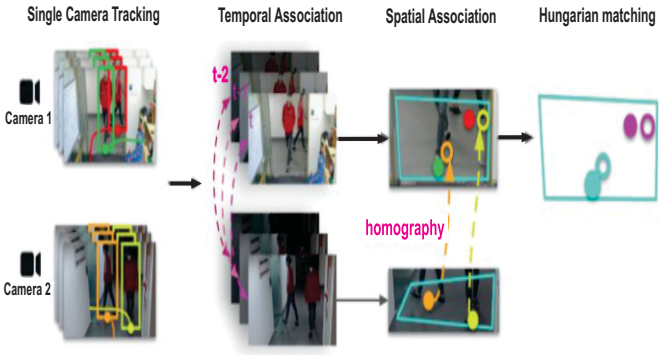


Fig. 14. Mockup of potential spatio-temporal heatmap of stray activity.

F. Unique Contributions

Our work introduces several novel contributions to the field:

- Persistent stray dog identification using YOLOv8m + ResNet50 — rarely applied together
- Scalable web deployment on Hugging Face for public interaction
- Real-time UUID tracking with transparent JSON logs and live UI feedback

This work goes beyond conventional one-time object detection by enabling civic authorities, NGOs, and communities to track stray dog identities longitudinally and take data-driven action.

VI. CONCLUSION AND FUTURE WORK

This paper presents a novel, end-to-end solution for the real-time detection and persistent re-identification of stray dogs in urban environments. Leveraging the powerful detection capabilities of YOLOv8m and the deep feature encoding of ResNet50, our pipeline not only detects stray dogs accurately but also assigns unique identities using cosine similarity–based matching. Each dog receives a UUID that is stored persistently across sessions, enabling accurate tracking, logging, and behavioral analysis.

By integrating this system with webcam input and a public Gradio web interface hosted on Hugging Face, we successfully enabled live deployment with minimal hardware requirements. This makes the solution viable for use by city authorities, NGOs, animal welfare groups, and local volunteers.

A. Summary of Contributions

- Developed a robust YOLOv8m-based dog detector fine-tuned on a hybrid dataset
- Designed a ResNet50-based Re-ID module for identity persistence
- Introduced a real-time UUID assignment and JSON logging framework
- Deployed the full system for public access via Hugging Face Spaces
- Enabled potential use cases for municipal stray control, pet recovery, and vaccination tracking

B. Proposed Future Enhancements

While our system is practical and scalable, several avenues for future research remain:

1) **Domain-Specific Re-ID Training:** Fine-tuning the ResNet50 feature extractor on a labeled stray dog dataset using Triplet Loss or Contrastive Loss could drastically improve re-identification accuracy, especially in visually similar cases.

2) **Dynamic Similarity Thresholding:** Rather than relying on a static cosine similarit{Citation}y threshold (e.g., 0.75), machine learning models could learn optimal identity matching boundaries per deployment region.

3) **Spatio-Temporal Tracking Across Cameras:** Integrating GPS-tagged multi-camera feeds, potentially building on lightweight multi-object tracking libraries [7], would enable re-identification across regions and heatmaps of dog movement patterns.

4) **Edge Deployment:** Using lighter models (e.g., MobileNet) and converting the pipeline to ONNX or TensorRT would allow deployment on mobile phones, Jetson Nano, or TPUs in field conditions.

5) **Behavior-Based Alert System:** Adding pose detection or motion trajectory analysis could help identify signs of aggression or injury, notifying authorities in real time and aiding intervention.

6) **Cloud Dashboard Integration:** A web-based admin interface built with Firebase or MongoDB can support querying, exporting, and visualizing re-identification logs, empowering NGOs to search UUIDs, detect return frequency, or predict high-risk regions.

7) **Citizen Participation via Tagging:** Future versions could allow citizens to upload dog sightings and tag them using mobile apps, creating a crowd-sourced dog database — a powerful feature for adoption drives and community monitoring.

As urban populations grow, stray animals increasingly intersect with human lives. This work takes a meaningful step in using AI for civic impact—enabling persistent, privacy-conscious, and intelligent tracking of stray dogs. With further refinement and collaboration, such systems can not only help manage urban wildlife but also build more compassionate, data-informed cities.

REFERENCES

[1] G. Jocher et al., "YOLO by Ultralytics," GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 770–778.

[3] J. Srinivas et al., "Dog Face Detection Using YOLO Network," International Journal of Advanced Research in Computer Science, vol. 10, no. 5, pp. 34–39, 2022.

[4] A. Das and R. Kumar, "Automated Detection and Threshold-Based Alerting for Stray Animals Using AI," in Proc. Intl. Conf. on Smart Cities, 2021.

[5] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in Proc. IEEE CVPR, 2015.

[6] A. Abid et al., "Gradio: Hassle-Free Sharing and Testing of ML Models in the Wild," in Proc. NeurIPS, 2019. [Online]. Available: https://gradio.app/

[7] P. Lorenzoni and A. Recasens, "Norfair: Lightweight and Flexible Object Tracking Library," 2021. [Online]. Available: https://github.com/tryolabs/norfair

[8] A. Sheikh and R. Siddiqui, "AI for Smart Cities: Applications and Ethical Considerations," AI & Society, vol. 36, no. 4, pp. 745–761, 2021.