**AI STRAY DOG DETECTOR USING MobileNetV2**

Submitted by: Karthikeyan S

Date: 10-05-2025

1. **Introduction**

Across both cities and villages, the rising number of stray dogs has become a significant concern—impacting public health, safety, and animal welfare. Unlike domestic dogs that receive care, shelter, and medical attention, stray dogs are often left to fend for themselves on the streets, lacking consistent access to food, safety, or healthcare.

This growing divide not only poses challenges for the animals but also creates difficulties for residents and adds pressure on local authorities striving to maintain public order and hygiene.

To address this issue, we introduce an **AI Stray Dog Detector** —an intelligent solution that uses **real-time video feeds** and **deep learning techniques** to distinguish between stray and domestic dogs. By leveraging the capabilities of computer vision, the system enables quick, accurate identification and classification of dogs in public spaces, supporting timely responses from authorities and helping create safer, cleaner communities.

**2. Objective**

The primary objective of this project is to develop an intelligent computer vision-based system that can detect, classify, and continuously monitor dogs in public spaces, with a focus on distinguishing between stray dogs and domestic (owned) dogs**.**

1. Automated Detection and Classification
2. Real-Time Monitoring and Data Logging
3. Support Public Safety and Policy Implementation
4. **Dataset Collection and Preparation**

3.1 Data Sources:

* To Collect manually images for stray dogs because there are not specific images to stray dogs
* There are more images for domestic dogs. We can download the images from the open-source platforms
  1. Labelling Strategy:

The label strategy uses a binary classification approach, where each dog is categorized as either stray or domestic.

* Stray dogs: Dogs with no visible owner, collar, or signs of domestication (Encoded as 1)
* Domestic dogs: Dogs that appear owned—e.g., with collar, clean appearance, or known to reside with humans (Encoded as 0)
  1. Preprocessing:

|  |  |
| --- | --- |
| Step | Description |
| Resize to 224×224 | Standardizes image size for MobileNetV2 input. |
| Dataset split | 80% training, 20% validation |
| Convert to array | Converts PIL image to NumPy array using img\_to\_array(). |
| Normalize pixel values | Uses preprocess\_input() from mobilenet\_v2 to scale pixels to [-1, 1]. |
| Expand dimensions | Adds batch dimension: (224, 224, 3) → (1, 224, 224, 3) for prediction. |
| Label encoding | Uses LabelBinarizer() to convert "stray\_dogs"/"not\_stray\_dogs" to 1/0. |
| One-hot encoding | Converts binary labels to [1, 0] and [0, 1] using to\_categorical(). |

Data Augmentation Type (Training Purpose):

|  |  |
| --- | --- |
| Augmentation Type | Parameter Settings |
| Rotation | rotation\_range=20 |
| Zoom | zoom\_range=0.15 |
| Width/Height Shift | width\_shift\_range=0.2, height\_shift\_range=0.2 |
| Shear | shear\_range=0.15 |
| Horizontal Flip | horizontal\_flip=True |
| Fill Mode | fill\_mode='nearest' |

1. **Model Selection and Training**

Model Used: **MobileNetV2**

Why MobileNetv2? Lightweight and Fast, Transfer Learning Friendly, Efficient Architecture, Balanced Performance, Compatibility

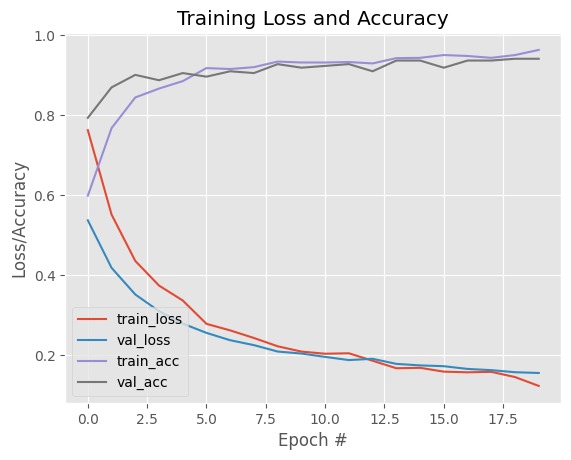
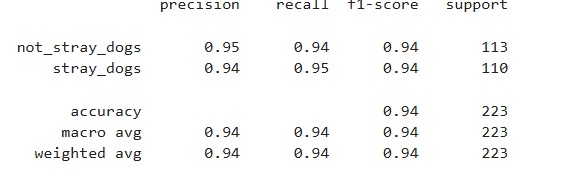
4.1 Transfer Learning Steps:

* Loaded pre-trained MobileNetV2 model
* Replaced classification head to output two classes
* Frozen base layers for initial epochs
* Fine-tuned with custom dataset

4.2 Training Environment:

* **TensorFlow / Keras** – for model building, training, and using MobileNetV2
* **OpenCV (cv2)** – for image and video processing, object detection
* **scikit-learn (sklearn)** – for label encoding and evaluation metrics
* **NumPy** – for numerical operations
* **Matplotlib** – for plotting training results
* **Epochs** – 20
* **Optimizer** – Adam

4.3 Performance Metrics:



1. **Results and Evaluation**

The model achieved strong accuracy (94%) in distinguishing between stray and domestic dogs, demonstrating reliable performance on both training and validation datasets.

Integrated with video feeds, the system accurately detected dogs in real time and correctly classified them, confirming its practical utility in public surveillance scenarios.

1. **Video Demonstration**

The generated output videos clearly display bounding boxes with class labels and confidence scores, providing immediate, understandable feedback for users and authorities.

1. **Future Scope**
2. Expand the dataset for identification of stray dogs classes
3. Extremely stray dogs are identified and trap them to cure or some other function (adapt, solve the problem of them, etc…,)
4. Allert for the government to rescue them
5. Continuous monitoring application development for stray dogs
6. **Conclusion**

The AI Stray Dog Detector using MobileNetV2 presents an innovative and practical solution to a growing urban and rural issue. By combining deep learning and real-time video analysis, this system offers a reliable way to distinguish between stray and domestic dogs—helping authorities and communities act more swiftly and effectively. With accurate detection, ease of deployment, and the potential for continuous improvement, this project lays the foundation for smarter, safer public spaces while also contributing to better animal welfare.

**Tools & Technologies Used**

|  |  |
| --- | --- |
| Category | Tools/Technologies |
| Programming Language | Python |
| Libraries & Frameworks | TensorFlow / Keras – for model training and transfer learning  OpenCV – for image and video processing  scikit-learn – for label encoding and evaluation  NumPy – for numerical operations  Matplotlib – for training visualization |
| Pretrained Model | MobileNetV2 |

**Keywords:**

Stray Dog Detection, MobileNetV2, Object Detection, Computer Vision, AI for Good.