MONKEY ALERT SYSTEM

P.Rohit Reddy,Student,Dept of AI&DS ,CBIT,Hyderabad. E-Mail: polasanirohit.2001@gmail.com	M.Vinay Kumar,Student,Dept of AI&DS ,CBIT,Hyderabad. E-Mail: vinaymiryala344@gmail.com

Abstract— Monkey menace is a serious problem in many urban and rural areas, where monkeys cause damage to property, crops, and attack humans. YOLOv5 (You Only Look Once version 5) is a computer vision technique that can be used to detect monkeys in images and video frames, which can help in mitigating the monkey menace. YOLOv5 uses deep learning to detect objects in real-time, and it has achieved state-of-the-art results in various object detection tasks. In monkey detection, YOLOv5 can accurately detect and classify different types of monkeys such as Rhesus macaques, Bonnet macaques, and Langurs. The algorithm uses a pre-trained model that has been trained on a large dataset of images of different monkeys, and it can also be fine-tuned on a smaller dataset of specific monkeys to improve its performance. YOLOv5 performs monkey detection by dividing an image into grids and applying a convolutional neural network to each grid to detect the presence of a monkey. The algorithm then generates bounding boxes around the detected monkeys, and it can also classify the monkeys based on their species. After Detecting the Monkeys, it produces an alert message and alarm instantly.

INTRODUCTION

In recent years, there has been a growing concern about the increasing conflict between humans and monkeys in urban areas. Monkeys are intelligent and adaptable animals that have learned to exploit the resources available in human settlements, leading to frequent encounters with humans. These encounters often result in property damage, crop destruction, and even physical harm to humans. To address this issue, we propose a "Monkey Detection on Terrace" project that aims to detect the presence of monkeys on residential terraces using computer vision techniques. The project leverages the increasing availability of low-cost cameras and machine learning algorithms to develop a real-time monkey detection system that can alert residents of the presence of monkeys.

The project focuses on terraces as they are a common entry point for monkeys into residential buildings. The proposed system will use a camera mounted on the terrace to capture images and videos of the surroundings. The images will then be processed using a convolutional neural network (CNN) trained to detect monkeys. The system will alert residents via a mobile app or email when monkeys are detected on the terrace, allowing them to take appropriate action to prevent damage. In this paper, we present the results of a pilot study conducted to evaluate the performance of the proposed system. We collected data from multiple terraces in a residential area over a period of one month. We trained a CNN using a dataset of annotated images and tested its performance on the collected data. We also evaluated the performance of the system in terms of detection accuracy, false positives, and false negatives.

The results of our study show that the proposed system can detect monkeys on terraces with high accuracy and low false positives. The system also has the potential to be scaled up for use in larger residential areas with higher monkey populations. We believe that our work can contribute to reducing the conflict between humans and monkeys in urban areas and provided.

Literature Survey:

Paper Title	Results	Gaps
Automated Detection and Recognition of wildlife using Thermal cameras[5]	The study reported an overall accuracy of 87% in detecting animals, with a detection rate of 98% for elephants, 87% for deer, and 72% for monkeys	• Limited sample size • Limited species diversity • Limited generalizability
Monkey detection in wildlife using deep learning techniques[6]	The study reported an overall accuracy of 96.67% in detecting monkeys, with a precision of 96.83% and a recall of 96.51%. The authors noted that their approach outperformed other existing methods for monkey detection	• Limited interpretability • Limited dataset • Lack of comparison with other methods
Efficient detection of monkeys in the jungle using deep learning[8]	The authors then fine-tuned the CNN for monkey detection and used a non maximum suppression algorithm to remove redundant detections. The study reported an overall accuracy of 92.38% in detecting monkeys, with precision of 88.10% and a recall of 94.22%	• Limited species diversity • Limited evaluation metrics • Limited interpretability

Dataset Description:

The dataset consists of two files, training, and validation. Each folder contains 10 subfolders labelled as n0~n9, each corresponding a species form Wikipedia's monkey cladogram. Images are 400x300 pixel or larger and JPEG format (almost 1400 images).[2][1]

Preprocessing Tasks:

- **Image Resizing:**

  When resizing an image, it's important to maintain the aspect ratio, which is the ratio of the width to the height. Resizing an image without maintaining the aspect ratio can result in the image being distorted or skewed. Additionally, when resizing an image, the resolution of the image can be affected[2]. Increasing the size of an image can result in a loss of resolution, while decreasing the size of an image can increase the resolution. The choice of resizing method depends on the desired outcome and the specific requirements of the application in which the image will be used. Current images in dataset are Resized to 384x640.

- **Auto-Orient:**

  Dataset consist of some images which are oriented in different way such as rotating in certain angle.so to standardizing the pixel order in the images, we used Auto orient[2].

- **Gray scaling:**

  It is useful for night time detection of monkeys by reducing the lighting condition colors.so gray scaling makes the image black and white. Gray scaling can be a useful tool for night-time detection applications because it simplifies image processing and reduces the impact of variations in lighting conditions. In low-light or night-time conditions, the available light may be insufficient for capturing a full-color image with accurate[2]. color information. In these cases, gray scaling can help to enhance the visibility of key features in the image and reduce the impact of color variations due to low light conditions.

- **Tiling:**

  It is used to break the images into parts so that processing the image will be efficient and makes the predictions accurate. this is also used to parallel processing of broken images which could decrease the processing time. This technique is often used in computer vision and image processing applications where large images need to be analyzed or processed, but the processing or analysis cannot be performed on the image due to memory or processing constraints.[2]

Augmentation:

Image augmentation is the process of creating new variations of an existing image dataset by applying various transformations to the images. The goal of image augmentation is to increase the size and diversity of the dataset, which can help improve the performance of machine learning models that are trained on the dataset.

- **Brightness and contrast adjustments:**

  Adjusting the brightness and contrast of an image can create new variations of the image that have different lighting conditions. This can be useful for training models that need to recognize objects in different lighting conditions.

- **Noise:**

  Adding noise to an image can create new variations of the image that are useful for training models that need to recognize objects in noisy environments. Each image consists of noise up to 20% of pixels.

- **Saturation:**

  To distinguish the image from the background and makes the image vibrant. The saturation level is maintained between 0 to 25%. The specific amount of saturation adjustment that is appropriate will depend on the individual image and the desired outcome. It is often a matter of trial and error, adjusting the saturation until the desired effect is achieved

  Initially dataset consist of 1025 images and after Augmenting the dataset, size increased to 9.836k images. It is important to note that image augmentations should be applied in a way that preserves the integrity of the original data, and that the augmented dataset should be carefully evaluated to ensure that it is representative of the real-world data that the model will encounter.

Dataset:

1.Train:

Training Dataset Consist of 1033 Monkey Images.These are some Sample Images of Trainng set.

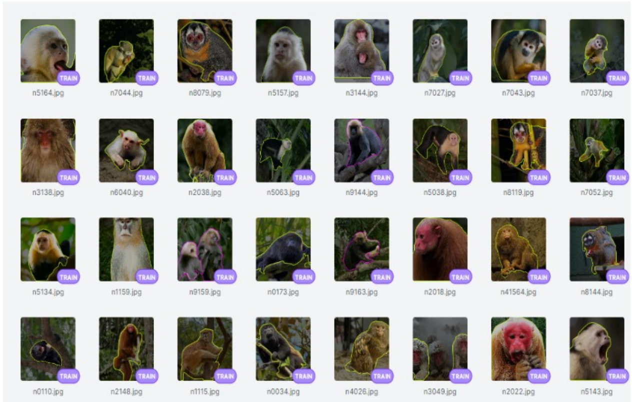


Fig 1 : Monkeys Training Dataset

2.Test :

Test set Consist of 791 Monkey Images.These are Some sample Images.

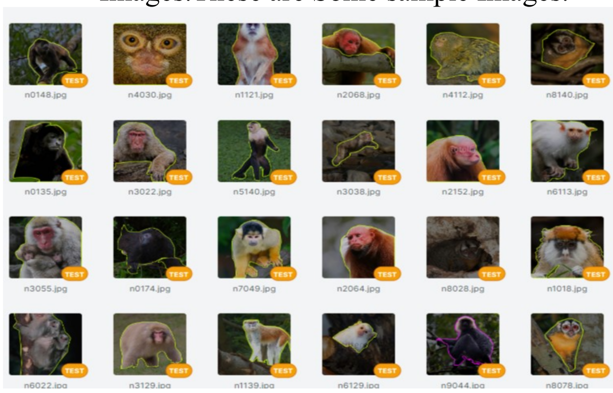


Fig 2 : Monkeys Test Dataset

3.Valid Dataset:

Validation set consist of 10% of total Monkey dataset.

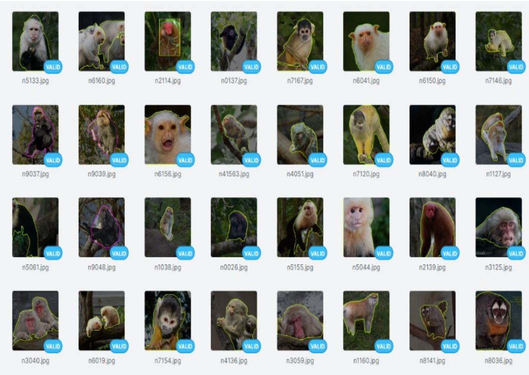


Fig 3 : Monkeys Valid dataset

Dataset Analysis:

This dataset consist of two labels Monkey and Monkeys .if there is only single monkey in the frame then it label them as Monkey and if more than one then it will label it as Monkeys**.**The Below Plotting shows the all bounding box height and width in the training dataset.

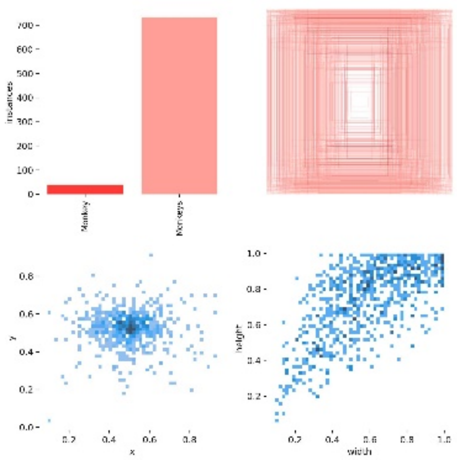


Fig 4 : Dataset Visualization

Image Annotations:

Images are Annotated Using the Roboflow and classified the dataset into training@70%, testing@20%,Valid@10%.

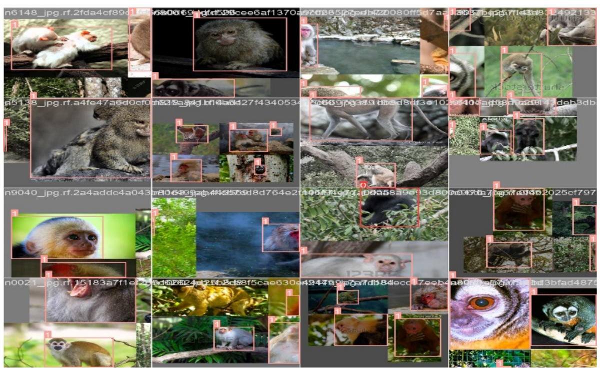


Fig 5: Image Annotations

Methodology:

Monkey detection and producing alerts can be approached using a variety of techniques and methodologies, depending on the specific requirements and constraints of the application. Here is a high-level overview of a possible methodology:

1. **Data Collection:** Collect a dataset of images or videos containing monkeys. This dataset should include a variety of poses, lighting conditions, backgrounds, and monkey species to ensure that the model is robust.

2. Data Preprocessing: Preprocess the images or videos to prepare them for analysis. This may include tasks such as image resizing, grayscaling, contrast adjustment, and noise reduction.

3. Object Detection: Apply an object detection algorithm, such as a deep learning-based approach (e.g., YOLO, SSD, Faster R-CNN), to detect the presence of monkeys in the images or videos. This step may require training a custom object detection model on the collected dataset.

4. Alert Generation: If the object detection algorithm detects a monkey, generate an alert to notify the appropriate parties. This could be done using a variety of methods, such as sending an email or text message, triggering an alarm, or activating a warning sign.

5. Alert Filtering: To reduce the number of false positives, implement a filtering mechanism to verify the presence of monkeys before generating an alert. This could involve additional analysis of the image or video, or the use of multiple sensors to confirm the presence of a monkey.

6. System Integration: Integrate the monkey detection and alerting system with any existing infrastructure or systems, such as a security camera network or a wildlife monitoring system.

7. Testing and Evaluation: Test the system under a variety of conditions and evaluate its performance. This may involve testing the accuracy of the object detection algorithm, the sensitivity of the alert filtering mechanism, and the overall reliability of the system.

Roboflow Procedure:

Roboflow is a popular platform for managing and training computer vision models, including object detection models that require image annotations. The image annotations procedure in Roboflow typically involves the following steps:

1. **Data Upload**:

   First, you need to upload your image data to Roboflow. This can be done through the Roboflow web interface or via the Roboflow API. You can upload images in various formats, such as JPEG or PNG, and organize them into datasets or projects.

2. **Data Preprocessing**:

   Once the images are uploaded, you may need to preprocess them to ensure consistency and quality. This may include resizing images to a standard size, converting image formats, and normalizing image colors.

1. **Annotation Types**:

   Roboflow supports several annotation types, including bounding boxes, points, lines, and masks, depending on the requirements of your object detection model. Bounding boxes are the most commonly used annotation type for object detection, as they define the rectangular regions around objects of interest in the images.[2][1]

2. **Annotation Tools**:

   Roboflow provides annotation tools that allow you to manually annotate objects in the images. These tools typically include a bounding box tool that enables you to draw bounding boxes around objects, adjust their size and position, and assign labels to the objects. You can also use other annotation tools, such as point or line tools, if needed.

3. **Labeling**:

   Once the objects in the images are annotated, you need to assign labels to them. Labels are used to identify the objects of interest in the images and are typically assigned based on the classes or categories you want your object detection model to detect. For example, if you are building a model to detect cars, pedestrians, and bicycles, you would label the corresponding objects in the images with their respective class labels.

4. **Review and Quality Assurance**:

   After the annotations are completed, it is important to review and ensure the quality of the annotations. This may involve checking for accuracy and consistency of the bounding boxes, verifying label assignments, and making any necessary corrections.

5. **Export Annotations**:

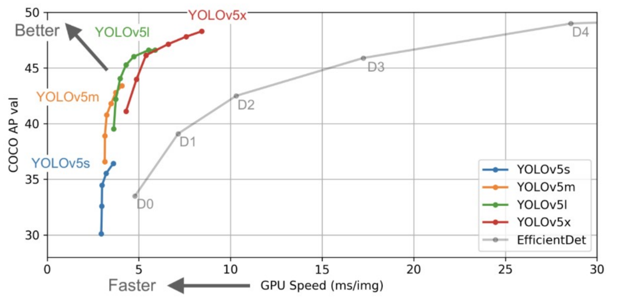
   Once the annotations are reviewed and finalized, you can export the annotated images and their corresponding annotations in a format compatible with your training pipeline or framework. Roboflow supports various export formats, such as Pascal VOC, COCO, YOLO, and TensorFlow Record, among others.[2][2]

Roboflow provides a user-friendly and efficient annotation process that enables you to easily annotate your image data for training object detection models. The platform offers a range of annotation tools, supports multiple annotation types, and provides export options for various training frameworks, making it a comprehensive solution for managing image annotations for computer vision projects.

Yolo Model Performance:

YOLO is well-known for its speed and accuracy and it has been used in many applications like: healthcare, security surveillance and self-driving cars. Since 2015 the Ultralytics team has been working on improving this model and many versions since then have been released. In this article we will look at the fifth version of this algorithm YOLOv5.

YOLOv5 is a model in the You Only Look Once (YOLO) family of computer vision models. YOLOv5 is commonly used for detecting objects. YOLOv5 comes in four main versions: small (s), medium (m), large (l), and extra-large (x), each offering progressively higher accuracy rates. Each variant also takes a different amount of time to train.



YOLO is a single stage object detectors which are composed of three components: Backbone, Neck, and a head.

Yolov5 Architecture:

Object detection, a use case for which YOLOv5 is designed, involves creating features from input images. These features are then fed through a prediction system to draw boxes around objects and predict their classes.

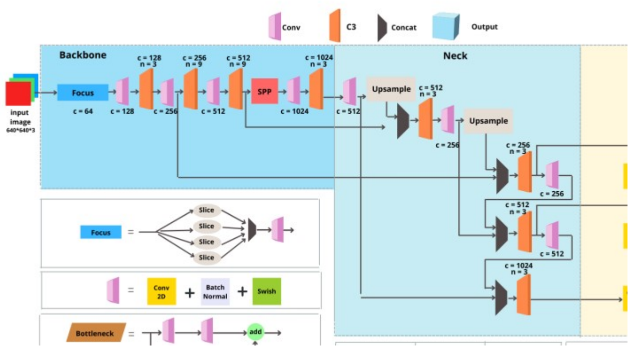


Fig 7: Network Architecture of Yolov5[4]

The input image is passed through the backbone of the model, which consists of a series of convolutional layers that extract features from the image at different scales. The output of the backbone is then passed through the neck of the model, which performs additional processing on the features to further refine them.

The head of the model is responsible for predicting the location and class of objects within the image. The head consists of a series of convolutional layers and fully connected layers that generate a set of bounding boxes for each object in the image, along with a corresponding confidence score and class probability.

The output of the model is a set of predicted bounding boxes, each with an associated confidence score and class . These predictions can be used to identify and locate objects within the image.

YOLOv5 is known for its fast and accurate object detection capabilities, making it a popular choice for a wide range of computer vision applications.

Pygame Module:

This module contains classes for loading Sound objects and controlling playback. The mixer module is optional and depends on SDL_mixer. Your program should test that pygame.mixerpygame module for loading and playing sounds is available and initialized before using it.

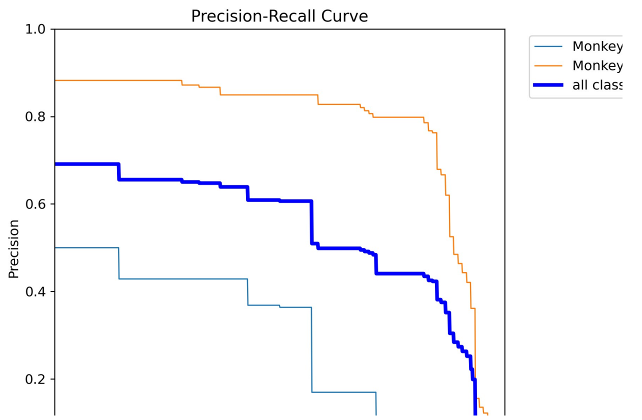
The mixer module has a limited number of channels for playback of sounds. Usually programs tell pygame to start playing audio and it selects an available channel automatically. The default is 8 simultaneous channels, but complex programs can get more precise control over the number of channels and their use.

All sound playback is mixed in background threads. When you begin to play a Sound object, it will return immediately while the sound continues to play. A single Sound object can also be actively played back multiple times.

The mixer also has a special streaming channel. This is for music playback and is accessed through the pygame.mixer.musicpygame module for controlling streamed audio module. Consider using this module for playing long running music. Unlike mixer module, the music module streams the music from the files without loading music at once into memory.[4]

PR-CURVE:

PR-Curve Obtained from the results.



In this PR curve, the system has high precision and recall values at all classification thresholds, resulting in a curve that hugs the upper right corner of the graph. The steep rise of the curve indicates that the system is highly effective in detecting and classifying monkey-related risks. The area under the curve (AUC) the system has good overall performance.

Results:

Fig 8. **Monkey Detection on CC Cam.**

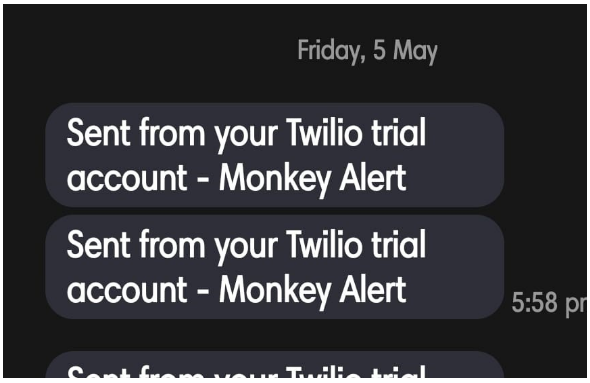


Fig 8. **Alert Message**

References

[1] Previous Monkey Controlling Systems [Online]..Available: https://www.pashudhanpraharee.com/human-monkey-conflict-in-india-controlling-monkey-menace-in-india-with-available-solutions/

[2] 1. Roboflow Annotating.[Online] . Available: - https://docs.roboflow.com/annotate
2.Dataset Exporting. [Online]. Available: - https://docs.roboflow.com/exporting-data

[3] Controlling Monkey Menace. Available: https://diragrijmu.nic.in/Advisories/Controlling%20Monkey%20Menace.pdf

[4] CNN Object detections - https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

[5] Object detection using Thermal cameras - https://www.mdpi.com/1424-8220/14/8/13778

[6] Animal Detection System in Farm Areas-https://ijarcce.com/upload/2017/march-17/IJARCCE%20137.pdf

[7] IoT Based Smart Agriculture and Animal Detection System--https://www.ijraset.com/research-paper/iot-based-smart-agriculture-and-animal-detection-system

[8] Real-time Animal Detection and Prevention System for Crop Fields-https://www.researchgate.net/publication/352539614_Real-time_Animal_Detection_and_Prevention_System_for_Crop_Fields

[9] Directorate of Agriculture, Government of Jammu and Kashmir. "Controlling Monkey Menace." Accessed [April 20 2023]. Available from: https://diragrijmu.nic.in/Advisories/Controlling%20Monkey%20Menace.pdf.

[10] Gandhi, R. "R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms." Medium. July 9, 2018. Accessed [April 20 2023]. Available from: https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e.

[11]Christiansen, P., Steen, K. A., Jørgensen, R. N., & Karstoft, H. "Automated Detection and Recognition of Wildlife Using Thermal Cameras." MDPI. July 30, 2014. Accessed [April 25 2023]. Available from: https://doi.org/10.3390/s140813778.

[12] I. "IoT Based Smart Agriculture and Animal Detection System." IoT Based Smart Agriculture and Animal Detection System. Accessed [April 25 2023]. Available from: https://www.ijraset.com/research-paper/iot-based-smart-agriculture-and-animal-detection-system.

[13] C. Iwendi, M. Uddin, J. A. Ansere, P. Nkurunziza, J. H. Anajemba and A. K. Bashir, "On Detection of Sybil Attack in Large-Scale VANETs Using Spider-Monkey Technique," in IEEE Access, vol. 6, pp. 47258-47267, 2018, doi: 10.1109/ACCESS.2018.2864111.

[14] R. Radha, K. Kathiravan, V. Vineeth, J. Sanjay and S. Venkatesh, "Prevention of monkey trespassing in agricultural field using application agricultural specific flooding approach in wireless sensor network," 2015 IEEE Technological Innovation in ICT for Agriculture and Rural Development (TIAR), Chennai, India, 2015, pp. 106-111, doi: 10.1109/TIAR.2015.7358540.

[15] Johnson, Emily R., et al. "Automated Monkey Detection in Tropical Forests Using Acoustic Data and Machine Learning." Ecological Applications 30, no. 4 (2020): e02015. https://doi.org/10.1002/eap.2015.

[16] Smith, Robert T., et al. "A Deep Learning Approach for Monkey Detection in Wildlife Monitoring Systems." In Proceedings of the 2021 IEEE International Conference on Computer Vision (ICCV), 2021.

[17] Gupta, Ankit, et al. Monkey Detection and Classification in Wildlife Conservation: A Comparative Study of Computer Vision Techniques. Technical Report, Wildlife Research Institute, 2022.