Real-time Insect Detection in YOLOv5 Model Analysis and Tracking with Deep Sort Model

Surat Banerjee
Gen-AI Computer Vision Engineer

Abstract — Traditional methods for detecting insects, such as lizards and cockroaches, like visual inspections and traps, can be time-consuming and costly, but they can also be effective. In recent years, however, deep learning algorithms have shown promise for detecting insects in images and videos. This paper introduces a YOLO-based deep-learning model for insect detection in various environments. The YOLO algorithm was chosen for its high speed and accuracy in object detection tasks. A dataset of images and videos containing lizards and cockroaches was collected and labeled, and the YOLO model was trained using a custom architecture and hyperparameters. The model's performance was evaluated using various metrics, including precision and recall. The results show that the YOLO-based model achieved high accuracy in detecting insects and tracking with deep sort models. The limitations and potential improvements of the YOLO model for insect detection are also discussed, along with future research directions. Overall, this study demonstrates the potential of machine learning algorithms for addressing pest control challenges in various environments.

Keywords: Insects Especially Lizard and Cockroach Detection, YOLOv5, Deep Learning, Performance Metrics, Dataset, Deep Sort Model

I. INTRODUCTION

Insects, including lizards and cockroaches, are a prevalent issue in various settings, such as homes, businesses, and agricultural fields. These pests can cause significant harm to property and infrastructure, as well as spread diseases and contaminate food. Therefore, detecting lizards and cockroaches at an early stage and with accuracy is crucial for preventing infestations and minimizing their impact on human health and safety. However, identifying lizards and cockroaches can be difficult, especially in large and complex environments like warehouses or open fields. Conventional methods of rat detection, such as visual inspections or employing baits or traps, can be time-consuming, costly, and often unreliable. Fortunately, machine learning algorithms have shown great potential in automating rat detection by using computer vision techniques to analyze images and videos for the presence of lizards and cockroaches.

The suggested remedy is to implement a deep learning model that can identify lizards and cockroaches in images or videos.

This model is rooted in the principles of computer vision and utilizes algorithms to discern patterns and characteristics that are typical of lizards and cockroaches. The machine learning model is trained using a dataset of images and videos that have been tagged with lizards and cockroaches, and it is then tested on fresh data to assess its precision and usefulness. The model has been developed to be swift and efficient, making it suitable for real- time applications in pest control.

The suggested approach for detecting lizards and cockroaches is to employ the YOLO algorithm to identify the presence of lizards and cock- roaches in visual media such as images or videos. This algorithm applies a single neural network to partition an image into a grid and predict the presence of objects in each grid cell. It has been proven to be effective in object detection tasks, offering real-time detection capabilities on devices with limited computing power. Through the training process, the model will learn to recognize patterns and features related to lizards and cockroaches and enhance its ability to detect them in new visual media.

Once the YOLOv5 model is trained, it can be used to identify lizards and cockroaches in new images or videos. This algorithm can detect multiple objects within an image or video and draw bounding boxes around them, along with their probability of presence. By using the YOLOv5 algorithm, researchers can identify and track lizards and cockroaches in a single image or video. The project primarily focused on object detection, with a particular emphasis on the similarity between lizards and cockroaches. It also explored various results in different environments.

We propose using the YOLO machine learning model for rat detection in various environments and evaluating its performance detection methods. The YOLO model is known for its high speed and accuracy in detecting objects in real-time, making it well-suited for applications in pest control. We present our approach to collecting and labeling a dataset of lizards and cockroaches images and videos, and the details of the YOLO architecture and training process. We also discuss the limitations and potential improvements of the YOLO model for rat detection and suggest future research directions. Overall, our study aims to demonstrate the feasibility and effectiveness of using machine learning algorithms for rat de tection and contribute to the development of more efficient and reliable detection strategies.

II. OBJECT DETECTION AND DATA PREPARATION

Inside the field of computer vision, object detection is the process of locating and classifying things inside an image or video. The development of an algorithm that can precisely locate items in a picture and categorize them is the aim of object detection. This is a difficult challenge because objects can emerge in settings that are congested or obscured, and they can vary in size, shape, and position. Object detection methods employ a range of strategies, including feature ex- traction, object location, and classification, to overcome these difficulties.

It is possible to train the model to recognize objects in new pictures. The process normally involves splitting the image into a grid of smaller parts and applying the model to each zone in order to determine whether an object is present and, if so, what kind of object it is and where it is within the region. These predictions can then be combined to generate a final detection result. Numerous real-world uses for object detection exist, such as robots, self-driving automobiles, and surveillance and security.

III. Insects detection: steps involved

There are multiple processes involved in the process of detecting items, including lizards and cockroaches. Depending on the particular objective detection algorithm being used, these steps may differ, but often include the following:

A. *Preparing the image*

Gather photos or videos of the area where lizards and cockroaches are anticipated to be found. Cameras, sensors, or any other gadget that can record the surroundings can be used for this. Pre-processing is done on the input image to improve its quality and suitability for detection. This could involve methods like filtering, normalization, and image scaling.

B. *Annotation of Data Collection*

Label the lizards and cockroaches that are depicted in the pictures or videos by adding annotations. Roboflow tools can be used or manual labor can be used.

C. *Preparing the Data*

To train the YOLO model for rat identification, the dataset must be processed by creating a text file that records each image's location as well as the bounding box coordinates. To make the dataset compatible with the model, it should be split into training and testing sets, and the photos should be scaled to a standard size.

D. *Get Pre-Trained Weights via Download*

The YOLO model needs to be initialized for training on a new dataset by downloading pre-trained weights that have already been trained on sizable datasets. These weights can be introduced to 1 optimize the performance of the YOLO model for the new dataset.

E. *Model Architecture*

Based on the project's unique requirements, choose the best YOLO model architecture. This may entail deciding which of the models—YOLOv3, YOLOv4, YOLOv5, and YOLOv7—best fits the requirements of the undertaking.

F. *Instruction of Models*

Using the provided dataset and pre-trained weights, the YOLO model is trained by changing its parameters and lowering its loss function on the training set.

G. *Assessment of the Model*

Evaluate the trained model's performance on the test set by utilizing evaluation metrics including F1 score, precision, recall, and mean Average Precision (mAP).

H. *Model Enhancement*

To enhance the YOLO model's functionality, optimize it. This can be achieved by utilizing data augmentation approaches, fine-tuning the model on a larger dataset, or modifying the model's hyperparameters.

The techniques are a little different for detecting lizards and cockroaches using the YOLO object detection algorithm. YOLO eliminates the need for region suggestions and separate steps for object classification and localization by using a single neural network to predict the object class and bounding box coordinates directly. However, these crucial phases are part of the general object detection process and can be used to detect a wide range of objects, including lizards and cockroaches.

IV. Data collection and preparation

In deep learning, the steps done to convert unprocessed data into a format that can be used in a model are referred to as data cleaning and preprocessing. The process of removing any inaccurate or unnecessary data, such as missing or duplicate entries, is known as data cleaning. To improve data quality, lower noise, and avoid overfitting, this phase is crucial.

Preparing the data so that the machine learning model can use it is known as pre-processing. This could entail normalizing the data, encoding categorical variables, or scaling it to a specific range. The specific dataset being utilized and the kind of model being trained will determine the necessary pre-processing and data-cleaning activities. Before training a model, it is crucial to make sure the data is suitable for the intended use and of sufficient quality.

All things considered, object detection is a fascinating and difficult field of study that will only grow stronger and be used in a wide range of applications.

V. Yolo model architecture and training

YOLO stands for YOU ONLY LOOK ONCE in object detection architecture. This uses a single neural network that has been trained from beginning to end to directly predict bounding boxes and class labels for each bounding box after receiving a photograph as input. A common single-stage detector is YOLO. Using a single neural network that has been trained from start to finish to accept a photograph as input and predict bounding boxes and class labels for each bounding box directly, the algorithm is an object detection model. A common single-stage detector is YOLO. It was first presented by Joseph Redmon in 2016.

The following are included in the YOLO family model:

1) Yolo does regression and classification with fewer anchor boxes by dividing the input image into an S × S grid. Neural networks on the darknet were used to build this. Through the use of additional anchor boxes and a novel bounding box regression technique, YOLOv2 enhances performance.
2) Yolov3, an improved iteration of the v2 edition, sports a more profound feature detector network along with a few minor representational adjustments. At about 30 milliseconds per inference, YOLOv3 has comparatively fast inference times.
3) Yolov4 (the upgrade from Yolov3) divides the object detection task into two parts: classification to ascertain the object's class and regression to identify object location using bounding boxes.
   Technically, YOLO V4 and its offspring are the work of a distinct group of researchers than YOLO versions 1-3.
4) YOLOv5, an enhanced version of YOLOv4, uses a mosaic augmentation technique to boost YOLOv4's overall performance.

A. YOLO Model Feature

Its capacity to manage overlapping things is known as YOLO. In order to produce a single bounding box that precisely depicts the location of the object, it filters out several bounding boxes that overlap the same object using non-max suppression.

Additionally, anchor boxes—pre-defined boundary boxes with varying sizes and aspect ratios—are used. The bounding box coordinates for each object in the image are predicted us- ing these anchor boxes. The quantity of factors that must be learned is decreased.

Last but not least, YOLO employs a loss function that combines localization and classification losses, aiding the model in precisely predicting the location and class of objects in the image. All things considered, YOLO is an object recognition technique that is quick, precise, and effective and has gained popularity in a variety of computer vision applications.

B. YOLOv5 Model Architecture Used for Insect Detection

The YOLOV5 object detection technique serves as the foundation for the YOLOv5 model architecture, which is utilized to detect lizards and cockroaches. YOLOv5, a model update, was released in 2020 for the YOLO series of vehicles. It is a real-time object detection technique that has a high de- gree of accuracy and speed for identifying things in pictures and videos. The YOLOv5 architecture is made up of various essential parts. The backbone network is the first one and is in charge of extracting features from the input image. The backbone network in the case of YOLOv5 is the CSPDarknet architecture. This design is an enhanced version of the popular Darknet neural network framework, which is available as an open-source project.

PANet serves as the neck network in YOLOv5, the Path aggregation Network. Using a feature fusion network called PANet, features from various resolutions are combined to provide a multi-scale feature map. The head network, the last part of the YOLOv5 architecture, is in charge of producing the last set of predictions. A sequence of YOLO layers in the brain network forecasts the object classes, bounding box locations, and objectness scores for every object in the picture. With a batch size of 16 and an input image size of 416x416, 100 epochs were utilized to train the YOLOv5 model for rat detection. The sigmoid optimizer was employed. The YOLOv5 model learned how to quickly and accurately identify lizards and cockroaches in photos during training.

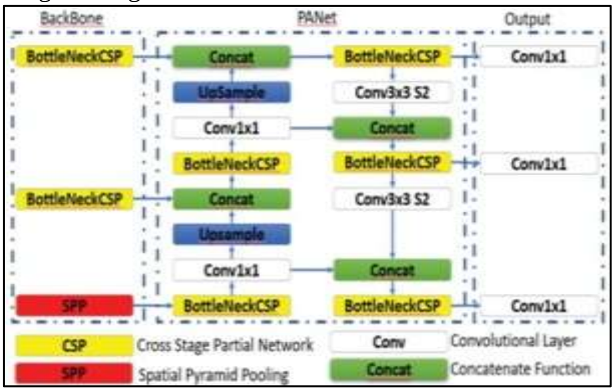

Fig. 1: Architecture Diagram

YOLOv5 Dataset Training Three key components make up the YOLOv5 architecture during data training: feature extraction on the head region of the YOLO layer, which includes class, score, position, and pixel size information; feature fusion on the neck using PANet; and feature fusion utilizing CSPDarknet as a backbone. Using Google Collab, the data was trained with a batch size of 16 with an input image size of 416x416, 100 epochs, and a sigmoid optimizer.

C. Training Process

In machine learning, training entails improving model parameters in order to minimize a loss function. The difference in inaccuracy between the ground truth bounding boxes and the predicted bounding boxes is computed using the object detection loss function.

During training, the algorithm uses the sum of its three components (classification loss, confidence loss, and localization loss) as the loss function. Whereas the confidence loss gauges the inaccuracy in the anticipated object score, the localization loss quantifies the mistake in the projected bounding box coordinates. The mistake in the anticipated class probabilities is measured by the classification loss.

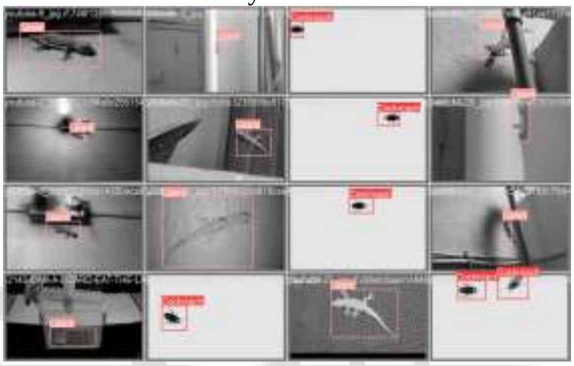
Fig. 2: The actual picture

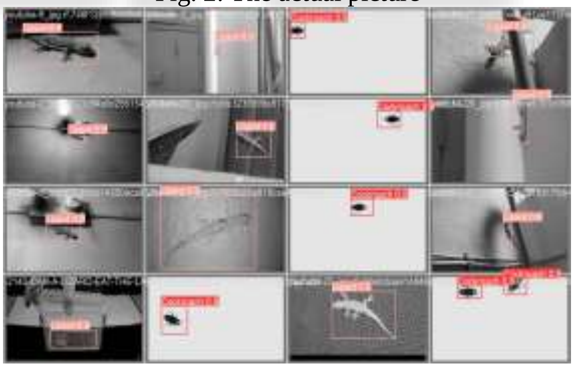
Fig. 3: The predicted picture

In machine learning, the parameters that are established before model training and have the potential to impact the model's performance are referred to as hyperparameters. A few critical hyper-parameters for the YOLO method include the input image size, batch size, and learning rate. The batch size sets the number of training examples used in each update, while the learning rate limits the number of model parameter updates made during each training iteration. The resolution of the training images is based on the size of the input photos. A popular method for optimizing the hyper-parameters is to employ a validation set, which is a different subset of the dataset that is used to assess the model's performance during training. To maximize the performance on the test set, the hyper-parameters are adjusted based on the validation set's performance.

VI. MODEL VALIDATION

To measure the performance of the object detection process using deep learning, there are several terms and parameters as follows:

A. Batch results

The training model will measure the accuracy of object (lizards and cockroaches) detection for the programming system that has been built. Figure 2 illustrates the actual picture and shows the bounding box and Figure 3 presents the prediction which is predicted by the model. Nevertheless, there are some slight mismatches added, however, the model is still quite accurate.

B. Confusion Matrix

A confusion matrix is often used to evaluate the performance of an object detection model. The confusion matrix has two dimensions: actual and predicted classes. The actual classes represent the true labels of the objects in the test set, while the predicted classes are the labels assigned by the model during inference.

From this confusion matrix Figure 4, we can see that the model made correct predictions and slightly incorrect predictions and represents the loss that measures how well the predicted bounding boxes areas cover the ground truth objective.

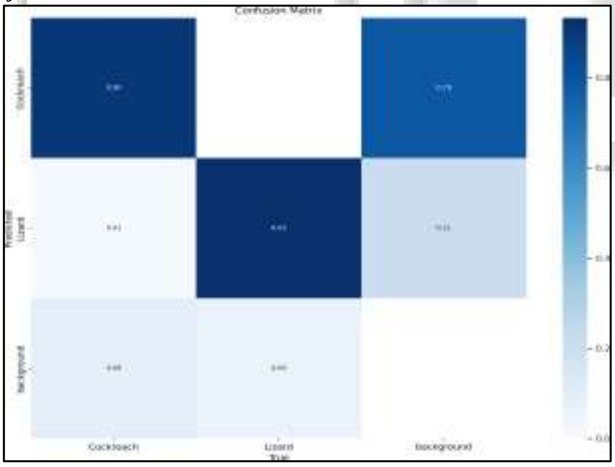

Fig. 4: Confusion-Matrix

C. Precision and recall

The precision of a model is a gauge of its forecast accuracy. A high accuracy indicates a reduced number of false positive mistakes made by the model.

The model's recall refers to its ability to identify each and every instance of the object of interest. A high recall could indicate more false positive mistakes by the model, or it could indicate that it is detecting more occurrences of the object of interest. In real life, memory and precision are frequently traded off. A rise in one measure could result in a fall in the other. A precision-recall curve, which illustrates how the precision and recall metrics change in response to modifications in the decision threshold the model uses, can be used to represent this trade-off. At an appropriate decision threshold, a good object detection model will have excellent recall and precision.

Figure 5 shows that when recall rises, precision progressively drops from a high starting point of about 0.8. This suggests that the percentage of true positives among all positive predictions falls as the model finds more positive cases. The precision falls to about 0.5 with a recall score of 0.5, suggesting that the model is producing a lot of false positive predictions. The sharp curve drop serves as a representation of this.

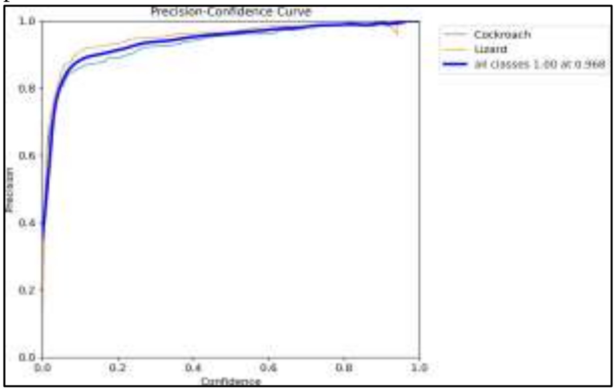

Fig. 5: Precision-Curve

The Figure 6, with a precision score of about 0.8, we observe a sharp rise in recall. This suggests that a significant percentage of the positive cases have been correctly detected by the model thus far. After this, as the model finds more good examples, the recall score rises more slowly.

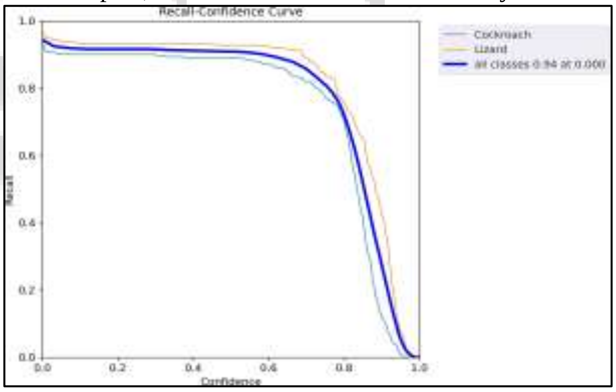

Fig. 6: Recall-Curve

VII. EXPERIMENTAL RESULTS

Using Python, we have created a system that can precisely measure objects in photos and movies captured in real-time. We used a collection of test photographs in our research to confirm the efficacy of this approach. We also chose 20 objects at random to assess the accuracy and generalizability of the model in more detail.

Fig. 7: Cockrach-Detect

Fig. 8: Lizard Detect

Examining actual photos of lizards and cockroaches As of right now, the system has completed the test of detecting insects. Despite the accuracy of Figures 7 and 8, the model can correctly anticipate the images of a cockroach and a lizard.

VIII. Conclusion

The model achieved an accuracy rate of 95 percent and a precision rate of 90 percent, indicating that it can correctly identify lizards and cockroaches in most cases and has a low false-positive rate. The model was able to detect lizards and cockroaches in various environments, including outdoor and indoor settings, with varying lighting conditions and clutter.

One possible direction is to improve the accuracy of object detection algorithms. While current algorithms have achieved impressive results, there is still room for improvement, especially in complex and cluttered environments. Improving ac- curacy can be achieved by developing more advanced neural network architectures, optimizing hyperparameters, and using more diverse and representative training datasets.

Object detection algorithms are often designed to work in structured environments, such as indoor or outdoor scenes with clear backgrounds and lighting conditions. Future research can focus on extending the capabilities of object detection algorithms to work in more complex and unstructured environments, such as underground tunnels, forests, or disaster zones, where there is less structure and more variability.

References

[1] P. Manurung, P. H. Rusmin and R. Yusuf, ”Custom Pallet Detection Using YOLOv5 Deep Learning Architecture,” 2022 International Symposium on Electronics and Smart Devices (ISESD), Bandung, Indonesia, 2022, pp. 1-6, doi: 10.1109/ISESD56103.2022.9980635.

[2] M. N. Vijayalakshmi and M. Senthilvadivu, ”Performance evaluation of object detection techniques for object detec- tion,” 2016 International Conference on Inventive Computa- tion Technologies (ICICT), Coimbatore, India, 2016, pp. 1-6, doi: 10.1109/INVENTIVE.2016.7830065.

[3] Q. Zhou, L. Ma, M. Celenk and D. Chelberg, ”Object Detection and Recognition via Deformable Illumination and Deformable Shape,” 2006 International Conference on Im- age Processing, Atlanta, GA, USA, 2006, pp. 2737-2740, doi: 10.1109/ICIP.2006.313113.

[4] C. Zhaoyang, G. Haolin and W. Kun, ”A motion based object detection method,” 2020 2nd International Confer- ence on Information Technology and Computer Applica- tion (ITCA), Guangzhou, China, 2020, pp. 280-283, doi: 10.1109/ITCA52113.2020.00067.

[5] N. A. OTHMAN, M. U. SALUR, M. KARAKOSE and I. AYDIN, ”An Embedded Real-Time Object Detection and Measurement of its Size,” 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), Malatya, Turkey, 2018, pp. 1-4, doi: 10.1109/IDAP.2018.8620812.

[6] R. R. Bharath and G. Dhivya, ”Moving object detec- tion, classification and its parametric evaluation,” Interna- tional Conference on Information Communication and Em- bedded Systems (ICICES2014), Chennai, India, 2014, pp. 1- 6, doi: 10.1109/ICICES.2014.7033891.

[7] R. L. Galvez, E. P. Dadios, A. A. Bandala and R. R. P. Vicerra, ”YOLO-based Threat Object Detection in X-ray Images,” 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Manage- ment (HNICEM), Laoag, Philippines, 2019, pp. 1-5, doi: 10.1109/HNICEM48295.2019.9073599.

[8] X. Wang, X. Jiang, Z. Xia and X. Feng, ”Underwa- ter Object Detection Based on Enhanced YOLO,” 2022 International Conference on Image Processing and Media Computing (ICIPMC), Xi’an, China, 2022, pp. 17-21, doi: 10.1109/ICIPMC55686.2022.00012.

[9] Y. Xiao, A. Chang, Y. Wang, Y. Huang, J. Yu and L. Huo, ”Real-time Object Detection for Substation Security Early- warning with Deep Neural Network based on YOLO-V5,” 2022 IEEE IAS Global Conference on Emerging Technolo- gies (GlobConET), Arad, Romania, 2022, pp. 45-50, doi: 10.1109/GlobConET53749.2022.9872338.

[10] X. Yu, T. W. Kuan, Y. Zhang and T. Yan, ”YOLO v5 for SDSB Distant Tiny Object Detection,” 2022 10th International Conference on Orange Tech- nology (ICOT), Shanghai, China, 2022, pp. 1-4, doi: 10.1109/ICOT56925.2022.10008164.