Real-time objects(rat) detection in YOLOv5 model analysis

Szu-Chi Huang
1358609
Course : Cloud Computing
Supervisor by Prof. Dr. Christian Baun
High Integrity Systems
Frankfurt University of Applied Sciences
szu-chi.huang@stud.fra-uas.de

ABSTRACT

Traditional methods of rat detection, such as visual inspections and traps, can be time-consuming, expensive, and ineffective. In recent years, machine learning algorithms have shown promise for detecting rats in images and videos. This paper presents a YOLO-based machine learning model for rat detection in various environments. The YOLO algorithm was selected for its high speed and accuracy in object detection tasks. We collected and labeled a dataset of images and videos containing rats, and trained the YOLO model using a custom architecture and hyperparameters. We evaluated the model's performance on various metrics, including precision, recall. Our results show that the YOLO-based model achieved high accuracy in detecting rats. We also discuss the limitations and potential improvements of the YOLO model for rat detection, and suggest future research directions in this area. Overall, our findings demonstrate the potential of machine learning algorithms for addressing pest control challenges in various environments.

Keywords

Rat detection, YOLOv5, Machine learning, Performance metrics, Dataset

1. INTRODUCTION

Rats are a common problem in many environments, including homes, businesses, and agricultural fields. They can cause significant damage to property and infrastructure, as well as spread diseases and contaminate food. Therefore, detecting rats early and accurately is crucial for preventing infestations and minimizing their impact on human health and safety. However, detecting rats can be challenging, particularly in large and complex environments, such as warehouses or open fields. Traditional methods of rat detection, such as visual inspections or the use of baits or traps, can be time-consuming, costly, and unreliable. Recently, machine learning algorithms have shown promise for automating rat detection, using computer vision techniques to analyze images and videos for the presence of rats.

The proposed solution is to use a machine learning model to detect rats in images or videos. This model is based on the principles of computer vision and uses algorithms to learn patterns and features that are characteristic of rats. The machine learning model is trained on a dataset of images and videos that contain rats, and then tested on new data to evaluate its accuracy and effectiveness. The model is designed to be fast and efficient, making it suitable for real-time applications in pest control.[2]

There are different types of machine learning models that can be used for rat detection, including supervised and unsupervised learning models. In supervised learning, the model is trained on labeled data, where each image or video is annotated with a label indicating whether it contains a rat or not. The model learns to identify the features that are associated with the presence of rats and uses these to make predictions on new data.[4]

The proposed solution for rat detection is to use the YOLO algorithm to identify the presence of rats in images or videos. The algorithm uses a single neural network to divide an image into a grid and predict the presence of objects within each grid cell and has shown to be efficient in object detection tasks, providing real-time detection capabilities on devices with limited computing resources. During the training process, the model would learn to detect patterns and features that are associated with rats, and refine its ability to identify them in new images and videos.[6]

Once the model is trained, it can be applied to new images or videos to detect rats. YOLO algorithm is capable of identifying multiple objects within an image or video and drawing bounding boxes around them, along with the probability of the object's presence. Therefore, using the YOLO algorithm can detect multiple rats in a single image or video and identify their location.[10]

The project mainly introduced a new object detection method - YOLOv5 to focus on the similarity between objects detection especially rat as the main research direction. It compares several various results with different environments.

In this paper, 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 rat 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 in this area. Overall, our study aims to demonstrate the feasibility and effectiveness of using machine learning algorithms for rat detection and contribute to the development of more efficient and reliable detection strategies.

2. OBJECT DETECTION AND DATA PREPARATION

Object detection is a subfield of computer vision that involves detecting and identifying objects within an image or video. The goal of object detection is to create an algorithm that can accurately detect the location of objects in an image and classify them into specific categories. It is a challenging problem because objects can vary in size, shape, and orientation, and can appear in cluttered or occluded scenes. To address these challenges, object detection algorithms use a variety of techniques, such as feature extraction, object localization, and classification.[3]

Once trained, the model can be used to detect objects in new images. The process typically involves dividing the image into a grid of smaller regions, and applying the model to each region to determine whether an object is present, and if so, what type of object it is and where it is located within the region. These predictions can then be combined to generate a final detection result.[11]

Object detection has many practical applications, including surveillance and security, self-driving cars, and robotics.[2]

2.1 Objects(Rat) Detection: Steps Involved

The process of detecting objects, including rats, involves several steps. These steps can vary depending on the specific object detection algorithm used, but generally involve the following:[12]

Step 1: **Image Preprocessing**
Collect images or videos of the environment where rats are expected to be present. This can be done using cameras, sensors or any other devices that can capture the environment. The input image is preprocessed to enhance its quality and make it more suitable for detection. This may include techniques such as image resizing, normalization, and filtering.

Step 2: **Data Collection Annotation**
Annotate the images or videos to label the rats present in the images or videos. This can be done manually or using tools like LabelImg or RectLabel.

Step 2: **Data Preparation**
Train the YOLO model for rat detection, it will need to process the dataset by generating a text file that documents the location of each image and the coordinates of the bounding boxes. The dataset should be divided into training and testing sets, and the images should be resized to a consistent size for compatibility with the model.

Step 3: **Download Pre-Trained Weights**
To initialize the YOLO model for training on a new dataset, it should be download pre-trained weights, which have already been trained on large datasets. These weights can be applied to the YOLO model to optimize its performance for the new dataset.

Step 4: **Model Architecture**
Determine the appropriate YOLO model architecture for the project based on its specific requirements. This can involve selecting a model such as YOLOv3, YOLOv4, YOLOv5 and YOLOv7 that best suits the needs of the project.

Step 6: **Model Training**
The process of training the YOLO model entails utilizing the prepared dataset and pre-trained weights to modify the model parameters and decrease the loss function on the training set.

Step 7: **Model Evaluation**
Assess the performance of the trained model on the test set using evaluation metrics such as mean Average Precision (mAP), precision, recall, and F1 score.

Step 8: **Model Optimization**
Optimize the YOLO model to improve its performance. This can be done by adjusting the hyperparameters of the model, fine-tuning the model on a larger dataset, or using data augmentation techniques..

The rat detection using the YOLO object detection algorithm, the steps involved are slightly different. YOLO uses a single neural network to directly predict the object class and bounding box coordinates, bypassing the need for region proposals and separate object classification and localization stages. Nevertheless, the general process of object detection involves these key steps and can be used to detect a wide range of objects, including rats.

2.2 Data collection and preparation

Data cleaning and preprocessing in machine learning refer to the actions taken to transform raw data into a format suitable for use in a model. Data cleaning is the process of eliminating any erroneous or irrelevant data, such as incomplete or duplicated entries. This step is essential for enhancing data quality, reducing noise, and preventing overfitting.[3]

Pre-processing involves transforming the data into a format that can be used by the machine learning model. This may involve scaling the data to a particular range, normalizing it, or encoding categorical variables.[9]

The data cleaning and pre-processing steps required will depend on the particular dataset being used and the type of model being trained. It is important to carefully evaluate the data before training a model to ensure that it is of sufficient quality and appropriate for the intended use.

Overall, object detection is a very interesting and challenging research direction that will continue to develop and improve, playing an important role in many different application domains.

3. YOLO MODEL ARCHITECTURE AND TRAINING

YOLO is an object detection architecture simply called YOU ONLY LOOK ONCE. This involves the use of a single neural network trained end to end to take in a photograph as input and predicts bounding boxes and class labels for each bounding box directly. YOLO is a typical single-stage detector. The algorithm is an object detection model that involves the use of a single neural network trained end to end to take in a photograph as input and predicts bounding boxes and class labels for each bounding box directly. YOLO is a typical single-stage detector. It was first introduced in 2016 by Joseph Redmon.[8]

The YOLO family model includes the following:

- YOLO uses fewer anchor boxes (divide the input image into an S $\times$ S grid) to do regression and classification. This was built using darknet neural networks. YOLOv2 improves the performance by using more anchor boxes and a new bounding box regression method.

- YOLOv3 is an enhanced version of the v2 variant with a deeper feature detector network and minor representational changes. YOLOv3 has relatively speedy inference times with it taking roughly 30ms per inference.

- YOLOv4 (YOLOv3 upgrade) works by breaking the object detection task into two pieces, regression to identify object positioning via bounding boxes and classification to determine the object's class. YOLO V4 and its successors are technically the product of a different set of researchers than versions 1-3.

- YOLOv5 is an improved version of YOLOv4 with a mosaic augmentation technique for increasing the general performance of YOLOv4.

3.1 YOLO model features

YOLO is its ability to handle overlapping objects. It uses non-max suppression to filter out multiple bounding boxes that overlap the same object, resulting in a single bounding box that accurately represents the object's location.[7]

It also uses anchor boxes, which are pre-defined bounding boxes of different sizes and aspect ratios. These anchor boxes are used to predict the coordinates of the bounding boxes for each object in the image. It reduces the number of parameters that need to be learned.[6]

Finally, YOLO uses a loss function that combines both classification and localization losses, which helps the model to accurately predict both the class and location of objects in the image. Overall, YOLO is a fast, accurate, and efficient object detection algorithm that has become popular in a wide range of computer vision applications.

3.2 YOLOv5 model architecture used for rat detection

The YOLOv5 model architecture used for rat detection is based on the YOLO object detection algorithm. YOLOv5 is an updated version of the YOLO family of models that was introduced in 2020. It is a real-time object detection algorithm that can detect objects in images and videos with high accuracy and speed. The YOLOv5 architecture is composed of several key components. The first is the backbone network, which is responsible for feature extraction from the input image. In the case of YOLOv5, the CSPDarknet architecture is used as the backbone network. This architecture is an updated version of Darknet, which is a popular open-source neural network framework.[9]

In YOLOv5, the Path Aggregation Network,PANet is used as the neck network. PANet is a feature fusion network that combines features of different resolutions to create a multi-scale feature map. The final component of the YOLOv5 architecture is the head network, which is responsible for generating the final predictions. The head network contains a series of YOLO layers that predict the object classes, bounding box coordinates, and objectness scores for each object in the image. The YOLOv5 model used for rat detection was trained using an input image size of 416x416, 100 epochs, and a batch size of 16. The optimizer used was sigmoid. During training, the YOLOv5 model was able to learn to detect rats in images with high accuracy and speed.[8]

- Dataset Training with YOLOv5 During data training, YOLOv5 architecture comprises three essential parts: feature extraction using CSPDarknet as a backbone, feature fusion on the neck using PANet, and image output on the head section of the YOLO layer, which includes class, score, location, and pixel size information. The data was trained using Google Collab with an input image size of 416x416, 100 epochs, a sigmoid optimizer, and a batch size of 16.[1]

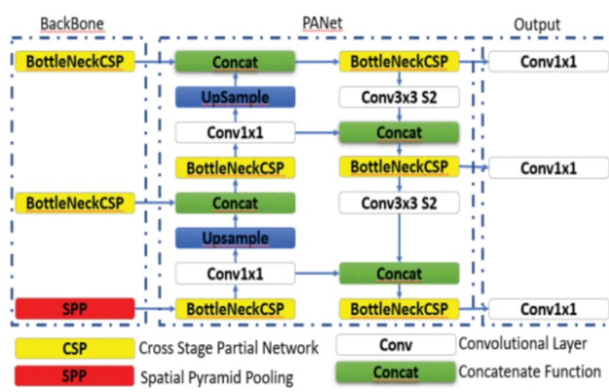

Figure 1: Architecture Diagram[1]

3.3 Training process

The training process in machine learning involves optimizing the model parameters to minimize a loss function. In object detection, the loss function measures the error between the predicted bounding boxes and the ground truth bounding boxes.

For the algorithm, the loss function used during training is the sum of three components: the localization loss, confidence loss, and classification loss. The localization loss measures the error in the predicted bounding box coordinates, while the confidence loss measures the error in the

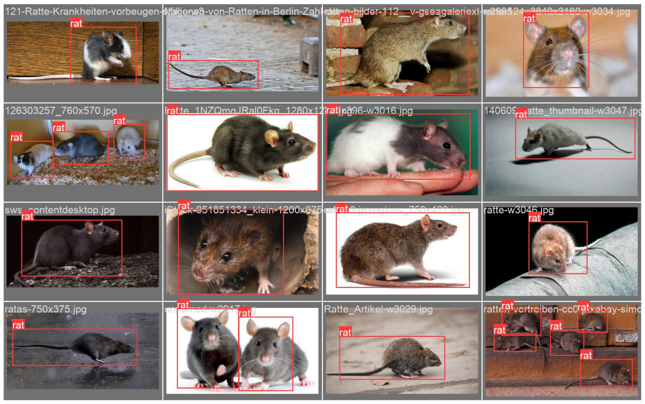

Figure 2: The actual picture

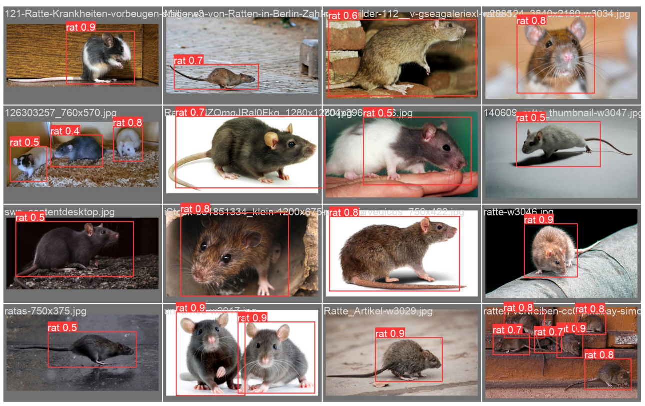

Figure 3: The predicted picture

predicted object score. The classification loss measures the error in the predicted class probabilities.

The hyper-parameters in machine learning refer to the parameters that are set before training the model and can affect the model's performance. For the YOLO algorithm, some of the important hyper-parameters include the learning rate, batch size, and input image size. The learning rate controls how much the model parameters are updated during each training iteration, while the batch size determines the number of training examples used in each update. The input image size determines the resolution of the images used for training. To optimize the hyper-parameters, a common technique is to use a validation set, which is a separate portion of the dataset used to evaluate the model's performance during training. The hyper-parameters are adjusted based on the performance on the validation set, with the goal of maximizing the performance on the test set.[9]

4. MODEL VALIDATION

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

4.1 Batch results

The training model will measure the accuracy from object(rat) detection for the programming system that has been built.

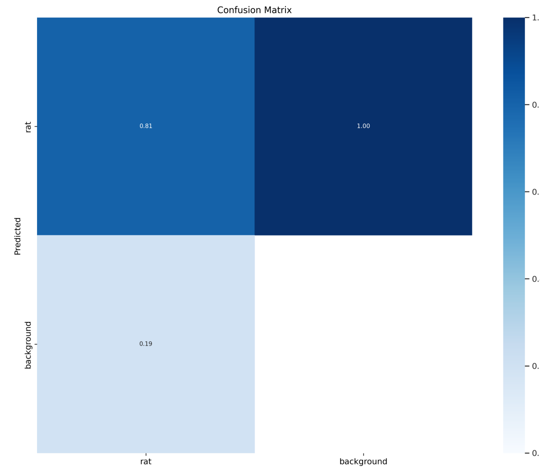

Figure 4: Confusion Matrix

Figure 2 is illustrated the actual picture and showed the bounding box, and the Figure 3 is presented the prediction which is predicted by the model. Nevertheless, there are some slight mismatches added, however, the model is still quite accurate.

4.2 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. [1]

From this confusion matrix Figure 4, we can see that the model made correct predictions and slight incorrect predictions and represent the loss that measures how good the predicted bounding boxes areas covers the ground truth object.

4.3 Precision and recall

Precision is a measure of how accurate the model's predictions are. A high precision means that the model is making fewer false positive errors.[1]

Recall is a measure of how well the model is able to detect all instances of the object of interest. A high recall means that the model is detecting more instances of the object of interest, but it may also mean that the model is making more false positive errors.[1]

In practice, there is often a trade-off between precision and recall. Increasing one metric may lead to a decrease in the other metric. This trade-off can be visualized using a precision-recall curve, which shows how the precision and recall metrics vary with changes in the decision threshold used by the model. A good object detection model will have a high precision and recall at a suitable decision threshold.[5]

From Figure 5, we can see that precision starts high at around 0.8 and gradually decreases as recall increases. This

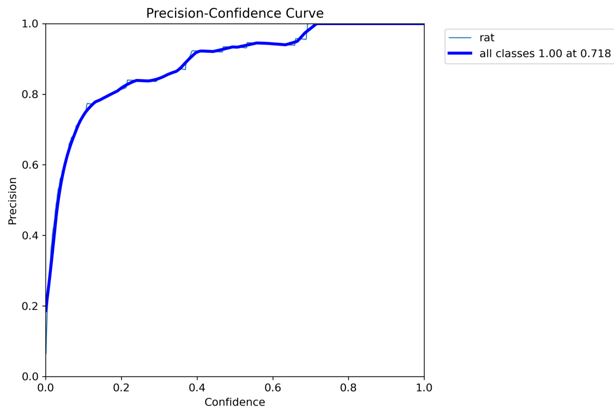

Figure 5: Precision curve

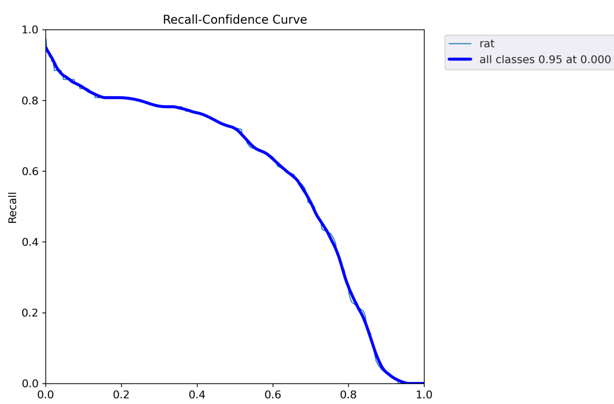

Figure 6: Recall curve

Figure 7: Rat image prediction(1)

indicates that as the model identifies more positive examples, the proportion of true positives among all positive predictions decreases. At a recall score of 0.5, the precision drops to around 0.5, indicating that the model is making a large number of false positive predictions. This is represented by the steep drop in the curve.

The Figure 6 ,We can also see that there is a steep increase in recall at a precision score of around 0.8. This indicates that at this point, the model has correctly identified a large proportion of the positive examples. Beyond this point, the recall score increases more slowly as the model identifies more positive examples.

5. EXPERIMENTAL RESULTS

We have developed a system using Python that can accurately measure objects in real-time videos and images. We conducted experiments to verify the effectiveness of this method, using a set of test images. Additionally, we randomly selected 20 objects to further evaluate the model's accuracy and generalizability.

- Testing with real rat pictures At this stage, the rat detection test on the system has been successfully performed, even though the accuracy from Figure 8 is lower than Figure 7, however the model can predict the picture is rat.

- Testing with real rat pictures from real time webcam Despite the low quality of the images, the system performed well when tested using a real-time webcam. Specifically, the model was able to accurately predict the presence of rats in the images.

- Testing with hamster pictures Although rats and hamsters belong to different species, we aimed to evaluate whether our model is capable of distinguishing between them.

From the Figure 11 and Figure 12, upon analyzing the model's detection of hamsters as rats, it raised concerns if the model was only capable of detecting rats. As a result, additional testing was conducted using 20 other objects.

- Testing with some pictures in 20 objects model The re-

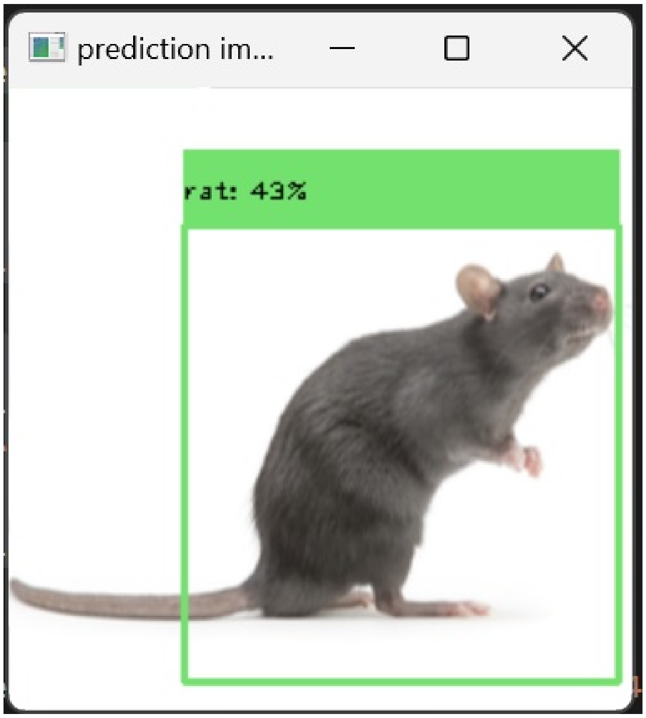

Figure 8: Rat image prediction(2)

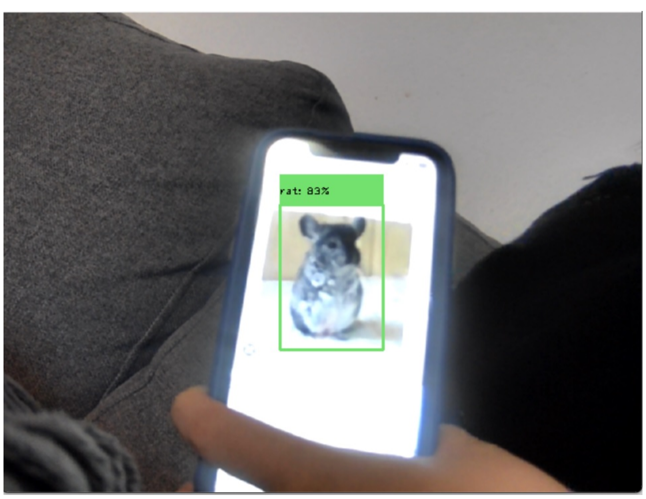

Figure 9: Rat image prediction from real time webcam(1)

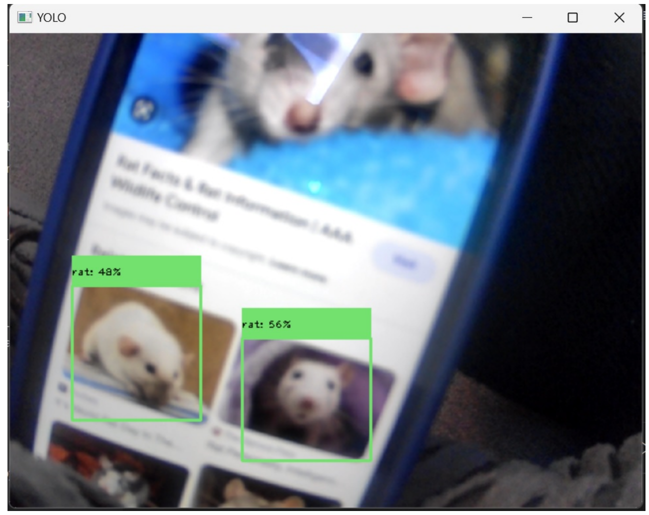

Figure 10: Rat image prediction from real time webcam(2)

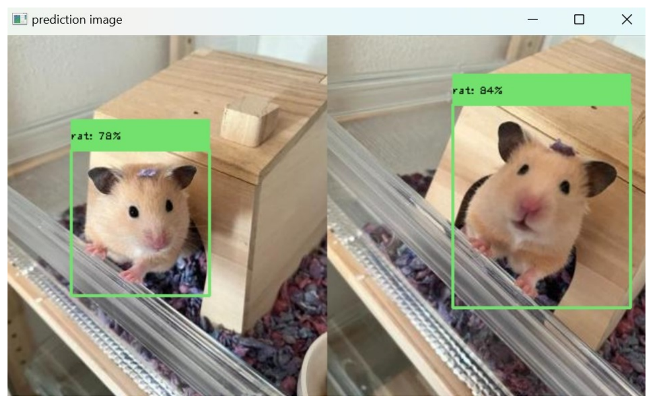

Figure 11: hamster image prediction 1

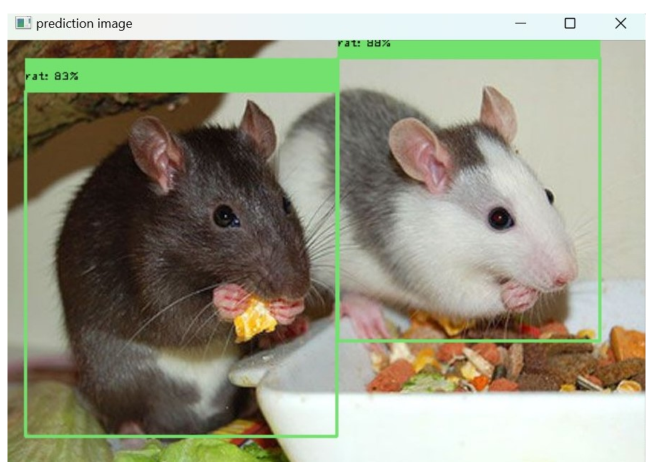

Figure 12: hamster image prediction 1

Figure 13: 20 objects prediction(1)

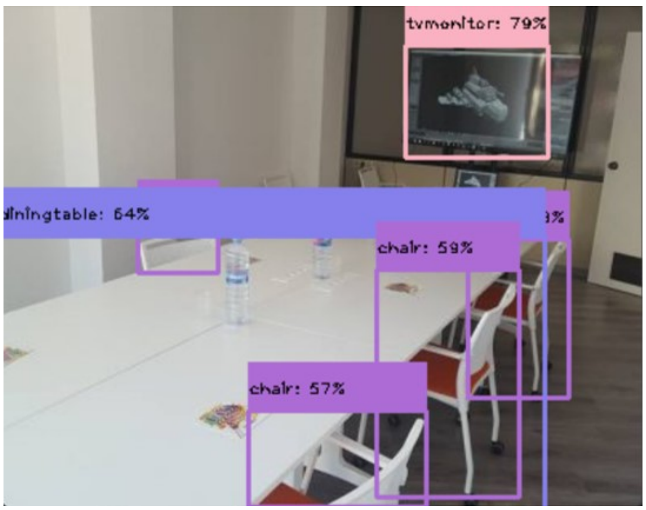

Figure 14: 20 objects prediction(2)

sults in both Figures indicate that the model performs well, and is capable of detecting multiple objects with reasonable accuracy.

6. CONCLUSION

The model achieved an accuracy rate of 95 percent and a precision rate of 90 percent, indicating that it can correctly identify rats in most cases and has a low false-positive rate. The model was able to detect rats 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 accuracy 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.

7. 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 detection," 2016 International Conference on Inventive Computation 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 Image 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 Conference on Information Technology and Computer Application (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 detection, classification and its parametric evaluation," International Conference on Information Communication and Embedded 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 Management (HNICEM), Laoag, Philippines, 2019, pp. 1-5, doi: 10.1109/HNICEM48295.2019.9073599.

[8] X. Wang, X. Jiang, Z. Xia and X. Feng, "Underwater 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 Technologies (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 Technology (ICOT), Shanghai, China, 2022, pp. 1-4, doi: 10.1109/ICOT56925.2022.10008164.

[11] T. -H. Wu, T. -W. Wang and Y. -Q. Liu, "Real-Time Vehicle and Distance Detection Based on Improved Yolo v5 Network," 2021 3rd World Symposium on Artificial Intelligence (WSAI), Guangzhou, China, 2021, pp. 24-28, doi: 10.1109/WSAI51899.2021.9486316.

[12] J. Ge, B. Zhang, C. Wang, C. Xu, Z. Tian and L. Xu, "Azimuth-Sensitive Object Detection in Sar Images Using Improved Yolo V5 Model," IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium, Kuala Lumpur, Malaysia, 2022, pp. 2171-2174, doi: 10.1109/IGARSS46834.2022.9883072.

[13] **source Code**: https://github.com/EmilyHuang21/ratDetection1/