IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

Tiny Object and Edge Detection Model

TOEDM

Hariharan T, Karthikeyan S Menon, Rohith S

Student (Team Leader) , Student , Student
BCA with Specialization in Artificial Intelligence,
Rathinam College of Arts and Sciences, Coimbatore, India

Abstract : This work presents a tiny dust detection and inspection method using a vehicle structure model which can operate in a desk automatically and turns by detecting the corners and edges using sensors. The TOEDR is a Tiny Object and Edge Detection Robot that performs based on Deep learning. It can able to detect the tiny objects like Pen marks, Pen, Pencil and Snacks. The dataset for this model is created using COCO dataset in Roboflow. A YOLOX algorithm has been proposed to recognize the dust on the top of the table. In addition, the planner framework was proposed to TOEDR for accomplishing the Edge Detection model which only gets run when camera in the model detects the object that has been trained. The Edge Detection is created based on IoT. Using Proteus the model framework is placed. The experimental shows that The Tiny Object Detection module achieves an average of 91 percentage detection accuracy, which is more suitable for deploying the TOEDR model for purpose of multi tasking. Further, the future work for the model that is to be add new features like cleaning module, placing camera in the model that is connected by bluetooth to the model.

IndexTerms - **Tiny Object Detection; Edge Detection; deep learning; TOEDR model; Dataset; IoT**

INTRODUCTION

Due to long working hours, low wages, unwillingness to work as a cleaner, workspace shortage has been a constant problem dust cleaning in desk on colleges and schools maintenance tasks in recent times. Recently, many robotic platforms are designed for different cleaning application which include floor cleaning, facade cleaning, staircase cleaning and garden cleaning. However, these robot architectures could not be support desk cleaning and maintenance tasks. In this context, TOEDR (Tiny Object and Edge Detection Robot) can be viable candidate for this task.

Various techniques have been developed for Model to recognize the different class of tiny dust (Pen mark, Pen, Pencil and Snacks) and delivers the output which class that belongs too. Among them, computer vision-based techniques are widely used in Tiny Object Detection for recognizing the dust and delivers the output. Model also detects the edge and corners of the desk by using IR and Ultrasonic sensors. The Robot that takes multiple actions like Dust detection and Edge detection. In the model various techniques were implemented. The techniques that we are used
YOLOX, ROBOFLOW, IOT, PROTEUS and ARDUINO IDE. Using these techniques Multi-task model
was determined. Let us go through these techniques

Proposed System

Figure 1 shows the functional block diagram of proposed scheme. It comprised of three modules, includes YOLOX based Object detection framework, Edge detection and cleaning module. The detection model comprises of two parts -a feature extractor and a bounding box predictor. Here, the feature extractor extract the specialized features to clean the dust that are fitted in Robot. As a consequence, bounding box predictors predicts the dust in an image and distinguish the classes.

3.1 YOLOX Based Object Detection

The YOLO framework (You Only Look Once) deals with object detection in a different way. It takes the entire image in a single instance and predicts the bounding box coordinates and class probabilities for these boxes. The biggest advantage of using YOLO is its simple and not complicated – it's incredibly fast and can process 45 frames per second. YOLO also understands

generalized object representation. This is one of the best algorithms for object detection and has shown a comparatively similar performance to the R-CNN algorithms.

3.2 HOW YOLOX FRAMEWORK WORKS

- YOLOX first takes a Live video:

- The framework the divides the input image into grids (3*3grid)

- Image classification and localization are applied on each grid. YOLOX the predicts the bounding boxes and their corresponding class probabilities for object (if any are found). Suppose we have divided the image into a grid of size 3 X 3 and there are a total of 3 classes which we want the objects to be classified into. Let's say the classes are Pen mark, Pen and Snacks respectively. So, for each grid cell, the label y will be an eight dimensional vector:

y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

Since there is no object in this grid, pc will be zero and the y label for this grid will be:

YOLOX decides whether there actually is an object in the grid. In the image, there are two objects (two snacks), so YOLOX will take the mid point of this two objects and these objects will be assigned to the grid which contains the mid-point of these objects. The y label for the centre left grid with the snack will be:

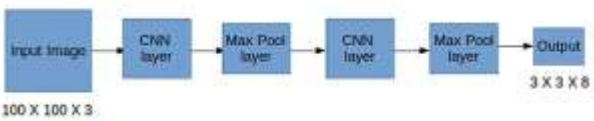
y =	0
	?
	?
	?
	?
	?
	?
	?

3.3 BOUNDING BOXES

The bx, by, bh, and bw are calculated relative to the grid cell we are dealing with. Consider the center-right grid which contains a pen mark:

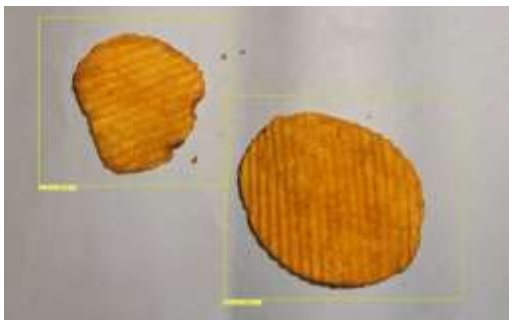
y =	1
	bx
	by
	bh
	bw
	0
	1
	0

So, bx, by, bh, and bw will be calculated relative to this grid only. The y label for this grid will be:



The bx, by are the x and y coordinates of the midpoint of the object with respect to this grid. In this case, it will be (around) bx = 0.4 and by = 0.3:

Now, bh is the ratio of the height of the bounding box (red box in the above example) to the height of the corresponding grid cell, which in our case is around 0.9. So, bh = 0.9. bw is the ratio of the width of the bounding box to the width of the grid cell. So, bw = 0.5 (approximately). The y label for this grid will be:



The bx and by will always range between 0 and 1 as the midpoint will always lie within the grid. Whereas bh and bw can be more than 1 in case the dimensions of the bounding box are more than the dimension of the grid.

3.4 EDGE DETECTION

Edge detection is one of the most important parts in the image processing systems of Desk Cleaning. It allows the extraction and display of feature such as curves, lines and angles for the purpose of identifying images from the images from the environment. Images represented by edges are widely used to simplify the process of identifying and interpreting images to the navigation system of cleaning robot which use camera as Primary sensor.

The main components used for Edge Detection:

- Arduino UNO
- Node MCU
- IR Sensor
- Ultrasonic Sensor
- Battery 9V
- Servo Motor
- BO Motor

PLANNER

The planner module is developed for accomplishing the table cleaning tasks through HSR. The planner has two function, namely finding the cleaning method and constructing the cleaning path. The process of flow diagram here as follows. It uses the dust cleaning framework for finding the cleaning method and constructing the cleaning path.

Two cleaning methods are adapted for the desk cleaning task, which include sweeping and wiping, where the sweeping method is used to clean the snacks and wiping method is used to clean the pen marks. Further, zig-zag cleaning action is consider for wiping the strains.

EXPERIMENTAL SETUP

This section describes the experimental setup of the proposed scheme. There are two phases in this experiment which are Object Detection and Camera Setup

3.1 Object Detection

- **Hardware**

You will need a robot platform, such as a mobile robot or a manipulator, that is capable of moving around the desk and cleaning it. You may also need additional hardware components, such as cameras, sensors, or actuators, to implement the desired functionality.

- **Software**

You will need to develop software for the robot, including a deep learning model for object detection and a control system for the robot. The deep learning model can be trained on a large dataset of images of desks, and the control system can use the outputs of the model to direct the robot to clean the desk.

- **Environment**

You will need a suitable environment to test your robot, such as a room or a lab with a desk. You should consider factors such as lighting, background, and clutter when selecting an environment to ensure that the robot will perform well in a realistic scenario

- **Data Collection**

You will need to collect a dataset of images of desks to train your deep learning model. This dataset should include a variety of different desk configurations and orientations, as well as different lighting conditions and backgrounds

- **Model Training**

You will use the collected data to train your deep learning model using a suitable deep learning framework, such as TensorFlow or PyTorch. You should evaluate the performance of your model on a separate dataset to ensure it is working as expected

- **Testing**

You will conduct a series of experiments to evaluate the performance of your robot, including its ability to detect and clean desks, its speed and efficiency, and its ability to handle various environmental conditions. You should document your results and analyse the performance of your robot to identify any areas for improvement.

3.2 DEPLOYMENT PROCESS

- **Overview of the process**

The desk detection and desk cleaning robot using deep learning is a system that combines the capabilities of deep learning algorithms and robotics to perform the tasks of detecting objects on a desk and cleaning them. The robot uses computer vision techniques to detect and classify objects on the desk, and then employs robotic actuators and grippers to physically interact with and clean the objects. The deep learning algorithms used in this system are trained on large datasets to recognize various objects and perform the cleaning task effectively. This type of system has the potential to significantly improve the efficiency and accuracy of desk cleaning and organization tasks,making it a valuable solution for businesses, homes, and other environments.

IV. RESULTS AND DISCUSSION

This section describes the Results and Discussions of the proposed scheme. There are three phases this section which including Tiny object detection, Edge Detection and Configuring both section.

4.1 Tiny Object Detection

In this session, Tiny Object Detection has accuracy of 91 percentage classes that find from Live camera.

4.2 Edge Detection

In this session, Edge Detection model detects the edges and corners by the help of sensors.

4.3 Configuring Both session

Configuring both session is done by using command prompt. The output and commands are listed below.

V. Conclusion

Developing a cleaning robot platform should have two critical features. The environment should be of real-world setting, and the platform should be able to reach every nook and corner of the cleaning region. The proposed framework is tested on a common college desk like setting so that the real-world implementation can be without any issues. Unlike the earlier results, the proposed work focuses on classifying the dust into various types, for which the cleaning process is different. The experimental results show that the proposed cleaning framework detects most of the Dust classes in desk with highest detection.

Developing a TOEDM platform should have two critical features. The environment should be of real-world setting, and the platform should be able to reach every nook and corner of the desk region. The proposed framework is tested on a common desk like setting so that the real-world implementation can be without any issues. Unlike the earlier results, the proposed work focuses on classifying the dust into various types, for which the classes process is different. Moreover, the classification module is augmented with an optimal path planning module and control module to achieve high efficiency.

I. Acknowledgment

We sincerely thanks to staff at Rathinam College of Arts and Sciences for their support.

REFERENCES

[1] Le, A.; Prabakaran, V.; Sivanantham, V.; Mohan, R. Modified a-star algorithm for efficient coverage path planning in tetris inspired self-reconfigurable robot with integrated laser sensor. *Sensors* **2018**, *18*, 2585. [CrossRef] [PubMed]

[2] Le, A.V.; Ku, P.C.; Than Tun, T.; Huu Khanh Nhan, N.; Shi, Y.; Mohan, R.E. Realization energy optimization of complete path planning in differential drive based self-reconfigurable floor cleaning robot. *Energies* **2018**, *12*, 1136. [CrossRef]

[3] Kouzehgar, M.; Tamilselvam, Y.K.; Heredia, M.V.; Elara, M.R. Self-reconfigurable façade-cleaning robot equipped with deep-learning-based crack detection based on convolutional neural networks. *Autom. Constr.* **2019**, *108*, 102959. [CrossRef]

[4] Ilyas, M.; Shi, Y.; Mohan, R.E.; Devarassu, M.; Kalimuthu, M. Design of sTetro: A Modular, Reconfigurable, and Autonomous Staircase Cleaning Robot. *J. Sens.* **2018**, *2018*, 8190802. [CrossRef]

International Research Journal
IJNRD
Research Through Innovation