APPLICATION OF FAST SEGMENT ANYTHING MODEL (FASTSAM) FOR AUTONOMOUS ROBOT IDENTIFYING PLANT DISEASE

Nguyen Thi Duyen
VietNam National University of Agriculture
Hanoi, Vietnam
ntduyen@vnua.edu.vn

Ngo Manh Tien
Institute of Physics VietNam Academy of Science and Technology
Hanoi, Vietnam
nmtien@iop.vast.vn

Nguyen Tran Hiep
Thanhdong University
Hai Duong, Vietnam
hiepnt@thanhdong.edu.vn

Ngo Quang Uoc
VietNam National University of Agriculture
Hanoi, Vietnam
ntduyen@vnua.edu.vn

Dam Quoc Vuong
HaNoi University of science and technology
Hanoi, Vietnam
thedqv@gmail.com

Do Quang Hiep
University of Economics-Technology for Industries
Hanoi, Vietnam
dqhiep@uneti.edu.vn

ABSTRACT

Large language models like Fast Segment Anything Model (FastSAM) have shown promising capabilities in few-shot learning across diverse domains. In this paper, we explore the application of FastSAM for plant disease identification by autonomous robots utilizing simultaneous localization and mapping (SLAM). We propose fine-tuning FastSAM on a dataset of plant images labelled with different disease types. The fine-tuned model is then deployed on an autonomous robot equipped with cameras and SLAM capabilities to identify plant diseases in real-world agricultural settings. Our results demonstrate that FastSAM can accurately recognize multiple plant diseases after being fine-tuned with only a few examples per class. The approach allows reliable plant disease identification without extensive training in data collection. This research highlights the potential of large language models like FastSAM for practical autonomous robot applications like precision agriculture when combined with technologies like SLAM.

Keywords: *Plant Disease, Segment Anything Model (SAM), FastSAM, Simultaneous Localization and Mapping (SLAM), Autonomous Robot.*

1. INTRODUCTION

Artificial intelligence has become indispensable for automating and optimizing various processes in smart agriculture [1]. One key application is automated plant disease identification, which is critical for timely disease control and minimizing crop losses. With the development of computer vision and machine learning, progress has been achieved in the recognition and diagnosis of plant diseases. There are many algorithms for detecting objects in images such as RCNN, Fast RCNN, Faster

RCNN and YOLO [2]. However recently, Meta AI Research introduced a novel image segmentation network with an inspiring name, Segment Anything (SA) [3]. Segment Anything Model (SAM) can segment any object within the image guided by various possible user interaction prompts. However, a real-time solution for the segment anything task, FastSAM [4] can solve multiple downstream tasks well.

Fast Segment Anything Model (FastSAM) is a recently proposed architecture for few-shot segmentation and detection of objects in images and videos with minimal training data requirements. FastSAM combines the strengths of the YOLOv8 object detector and the OpenAI CLIP image encoder for efficient and flexible few-shot learning. YOLOv8 [5], [6] is an optimized object detection model that builds on the YOLO family of detectors using a Transformer as its backbone. It achieves excellent performance on common object detection benchmarks while being fast and efficient. On the other hand, CLIP is a contrastive image-text model trained by OpenAI on a diverse dataset of image-caption pairs. It encodes images and text into a common embedding space allowing zero-shot inference by comparing image and text embeddings. FastSAM utilizes both these components in an innovative fashion for few-shot segmentation. The YOLOv8 object detector proposes candidate object regions while CLIP provides the segmentation masks. During training, CLIP's rich world knowledge helps FastSAM generalize better from few examples. At test time, FastSAM takes as input the target object name and few segmentation masks and can detect and segment all instances of the object. This enables segmenting new object categories with just 1-5 examples, hence the name few-shot segmentation.

Though originally proposed for generic objects, recent studies have explored the potential of SAM models for specific domains like medical imaging [7], information [8], and construction [9]. However, their application for agricultural tasks like plant disease identification remains relatively underexplored. Some recent works have studied semantic segmentation for leaf disease detection using models like DeepLab v3+. But dedicated investigations into leveraging SAM architectures like FastSAM for plant disease recognition are still lacking. Our work aims to address this gap by proposing and evaluating the use of FastSAM models for plant disease identification with minimal training data.

Simultaneous localization and mapping (SLAM) [10] is a key capability required by autonomous robots and unmanned ground vehicles (UGVs) for navigation in unstructured environments. SLAM enables a robot to incrementally build a map of the surroundings while

simultaneously localizing itself within the map. Robot Operating System (ROS) [11] provides robust open-source implementations of SLAM algorithms and frameworks for UGVs. This allows leveraging ROS-based SLAM capabilities for navigation of robots aimed at agricultural and field applications.

This paper proposes a Fast Segmentation Anything Model (FastSAM) based approach for enabling autonomous robots to accurately identify multiple plant diseases with minimal training data. The core FastSAM architecture uses a YOLOv8 object detector combined with an OpenAI CLIP encoder trained on disease-specific prompts. This model is fine-tuned on a small dataset of annotated plant images with multiple disease types. The fine-tuned FastSAM model is deployed on an autonomous robot equipped with cameras and SLAM capabilities via ROS to create a plant disease map. Our experimental results demonstrate that the proposed approach can reliably recognize several plant diseases after fine-tuning with just 50-100 annotated examples per class. The system achieves approximately 90% accuracy in classifying test disease images captured under real-field conditions. The proposed FastSAM-based framework demonstrates promising capabilities for practical autonomous plant disease identification with low data requirements.

2. METHOD AND MATERIALS

2.1 Collection dataset

The data set we used are images of cucumber plants with diseases and normal cucumbers collected from cucumber greenhouses (Fig.1). Photos were taken from phones and cameras with a resolution of 8MB or more. Our dataset consists of collecting 2000 images, dividing them into 80% training images and 20% testing images.

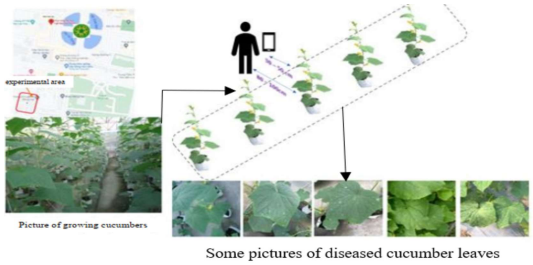

Fig. 1. *Data collection process [17]*

2.2 Fast Segmentation anything model (Fast SAM)

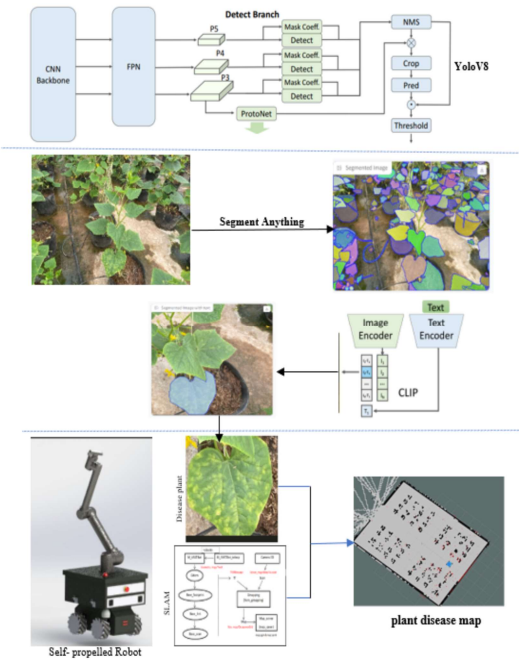

Fig. 2. *Overview: Training Plant Disease Image Segmentation with SLAM for Disease Mapping*

FastSAM is a two-stage framework comprising All-Instance Segmentation (AIS) and Prompt-Guided Selection (PGS). Fig. 2 gives the overview of the proposed method, FastSAM. First, we employ YOLOv8 [12] for segmenting all objects or regions in an image. Then, we leverage various prompts like point prompts, box prompts, and text prompts (based on CLIP [13]) to pinpoint the specific object(s) of interest.

2.2.1. Instance Segmentation

YOLOv8 employs YOLACT [14] principles for instance segmentation. It starts with feature extraction via a backbone network and Feature Pyramid Network (FPN) [12], incorporating features of various sizes. The output encompasses detection and segmentation branches.

Fig. 3 illustrates YOLOv8's architecture. YOLOv8 shares a similar backbone with YOLOv5 but includes modifications like the CSPBottleneck, now called the CSP-M module [12]. The CSP-M module amalgamates high-level features with contextual information to enhance detection accuracy.

YOLOv8: The Ultimate Object Detection Model

YOLOv8, the latest iteration of the YOLO family, excels in joint detection and segmentation. It boasts a revamped architecture, enhanced convolutional layers, and an advanced detection head, making it ideal for real-time object detection. YOLOv8 supports cutting-edge computer vision algorithms, including instance segmentation, and employs the CSPDarknet53 backbone network [15]. It adopts an anchor-free model with a decoupled head to independently process objectness, classification, and regression tasks, elevating accuracy. The model integrates Feature Pyramid Networks for recognizing objects of various sizes. YOLOv8 offers a user-friendly API and suits real-time detection tasks, especially in public safety and emergency response.

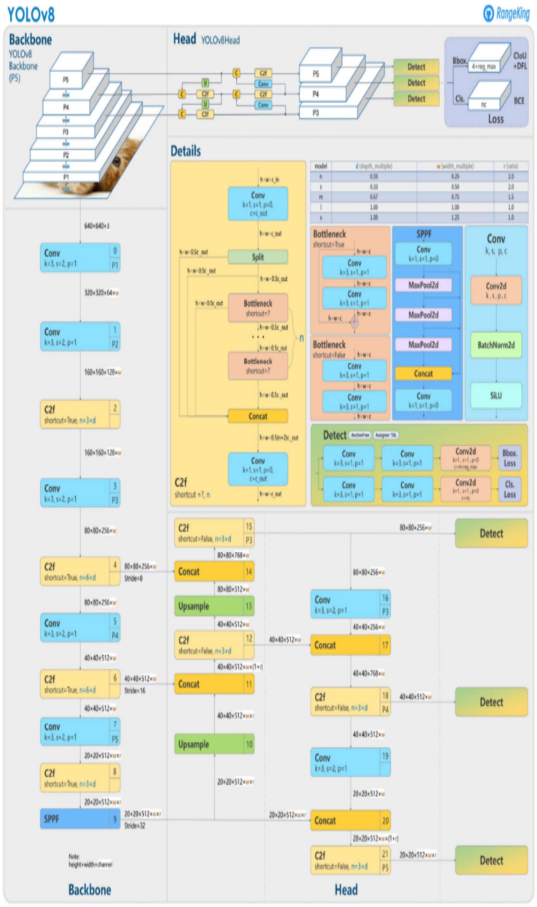

Fig. 3. *YOLOv8 Architecture [4]*

2.2.2. Text prompt

In the case of text prompts, we extract text embeddings using the CLIP [12] model. Subsequently, we determine image embeddings and match them with the intrinsic features of each mask using a similarity metric. The mask with the highest similarity score to the image embeddings of the text prompt is selected.

To simulate the prompt generation process, we first generate "ground prompts" based on the ground segmentation masks and then introduce some randomness. The pseudocode For each img_file in the directory at dataset_path:

- Set img_path to the path of img_file in dataset_path
- Load the image from img_path into the variable image
- Set mask_file to img_file with '.jpg' replaced by '_mask.png'
- Set mask_path to the path of mask_file in dataset_path
- Load the mask from mask_path into the variable mask
- Set text to randomly_select_text(mask)
- Set bbox_width to mask_width + 20
- Set bbox_height to mask_height + 20
- Set bbox_x to bbox_x + random integer between -30 and 30
- Set bbox_y to bbox_y + random integer between -30 and 30
- Multiply bbox_width by random float between 0.9 and 1.1
- Multiply bbox_height by random float between 0.9 and 1.1
- Overlay the bounding box and text on the image as prompts using overlay_bbox_and_text(image, bbox, text)

2.2.3. Data processing, labelling, and training

2.2.3.1. Data processing and labeling

The data set is divided into 3 groups: Normal cucumber group, Cucumber plant group with downy mildew disease, Cucumber plant group with powdery mildew disease.

We used the tool on Labelme to label each group of objects. After receiving the .json files converted to .txt files by the labelme2yolo tool to get the data set for the training process. The labelling process and description are illustrated in Figures 4 and 5.

Fig. 4. *Object labelling process diagram*

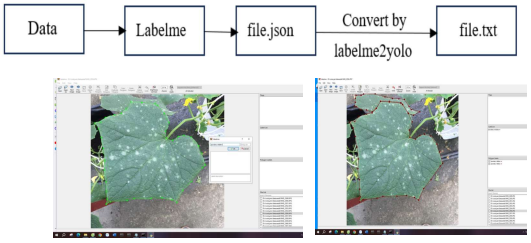

Fig. 5. *Description of the labeling process*

2.2.3.2. Training and testing datasets

The hold-out is a method of randomly dividing the data set into two parts, the first part is used for training and the second part is used for testing. Usually, the test data set is taken from the data set of about 10%-30%, and the remaining is the training data set (90%-70%). This paper, we divided the dataset into training (80%) and validation (20%) through random sampling for both YOLOv8 and fastSAM models.

If the results of the model are not satisfactory, we can proceed to refine the model as follows:

The image encoder employed the pre-trained fastSAM model. All image embeddings were computed offline by supplying the normalized images to the encoder, which then resized them to dimensions of 1024 × 1024 × 3. We derived the bounding box prompt from the ground-truth mask using the method we showed in the previous section. Our chosen loss function combined dice loss and cross-entropy loss without any weighting, a strategy that has demonstrated effectiveness in a wide array of segmentation tasks. To optimize the network, we used the Adam optimizer and set the initial learning rate to 1e-5. To decide when to stop training, we monitored the performance of the validation set. We ceased training when the model's performance plateaued, indicating that additional training would not yield significant improvements and could lead to overfitting.

The PC we used to finetune the model has the parameters given in the table 1. The training and inference were run on a 6GB NVIDIA GeForce RTX 1660 GPU, using YOLOv8.0.53, Python 3.9.17p, PyTorch 2.0.1 and CUDA 12.2, on openrating system windown 10.

Table 1. The parameters of pc

Computer Configuration	**Specific Parameters**
CPU	Intel XEON
GPU	GTX 1660
Operating system	Windown 10
Memory GPU	6 GB
Memory Ram	64 GB

2.3 Self-propelled Robot Model and SLAM technique

2.3.1 Self-propelled Robot Model

The structure of the autonomous robot monitoring greenhouse crops is shown in the figure 6.

Fig. 6. *Self- propelled Robot Model*

1- Cameras;
2- Manipulator
3- Central processing system and Power supply
4- Lidar
5- Self-propelled wheel system

2.3.2 SLAM technique

The SLAM processing technique will provide map information about the environment as well as estimate the robot's own posture (position and orientation) based on signals received from vision sensors including Rplidar and 2D camera.

Gmapping can perform well for a less processing power robot. The mapping package in ROS provides laser-based SLAM (Simultaneous Localization and Mapping), as the ROS node called slam_gmapping [16].

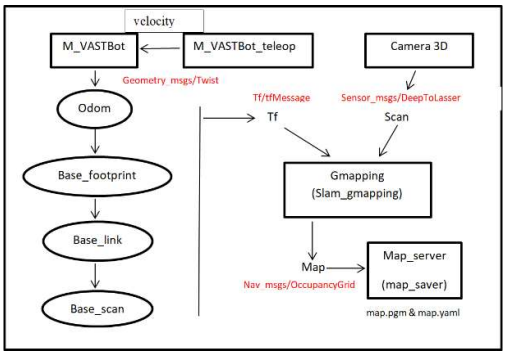

Fig. 7. *Gmapping package*

The Gmapping package estimates the Robot's position and builds a map based on the acquired data and its geometric measurements (Fig. 7) .

- The TF library is designed and provides standards for tracking coordinate frames and data transformation throughout the system.
- RPLidar: This node runs the RPLidar sensor and sends the "scan" necessary information for SLAM to the Gmapping node.
- Teleop: This node is the control algorithm so that the Robot can move according to the user's wishes. Then proceed to send angular velocity and moving speed to the core based on the received signal.
- Core: This node receives the moving velocity and angular velocity. While Publishes “odom”, it is the measured and estimated pose of the Robot. Besides, we also publishes the robot's coordinates which have been converted in the order:
- Odom→Base_footprint→Base_link→Base_scan.
- Then, these data will be sent to topIC tf.
- Gmapping: This node will create a map based on distance measurement information from the laser scan signal and information from topIC tf, which is the robot's posture.
- Map_server: This node creates the file "map.gpm" and the file "map.yaml", these two files contain the information of the obtained map.

3. RESULT AND DISCUSSION

3.1 Segment anything

We first visualize some segmentation results in Figure 8

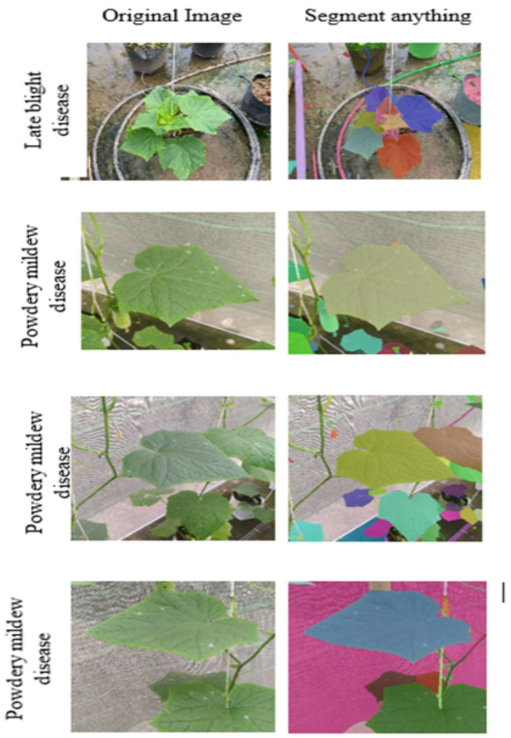

Fig. 8. *Segmentation Results of FastSAM*

3.2 Promt interaction (Text prompt) vs. No Prompt (Detection by using YOLOv8)

One of the main features of the fastSAM system is prompt interaction. We compared the performance of diseased leaves segmentation using text mode with text prompts and Yolov8 without prompts.

As shown in Figure 9, the segmentation results with prompt are highly satisfactory. In particular, to identify diseases on plants, fastSAM only requires a small data set (about 300 images).

Although, Yolov8 detected more aim objects, this model is trained on a much larger data set (over 2000 images).

This result shows that by using text mode in fastSAM, the robot provides information about leaves or leaf areas with the most obvious signs of pests and diseases to perform appropriate tasks.

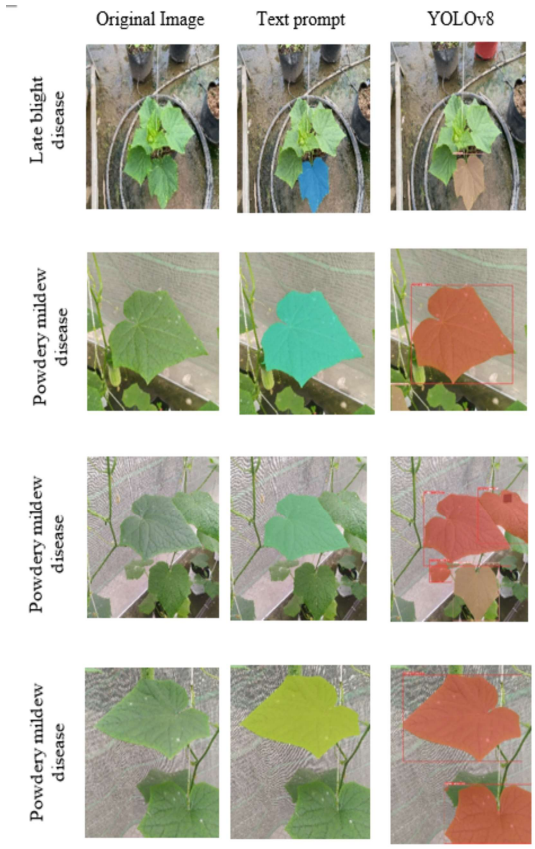

Fig. 9. *Images of segmentation of disease plants with and without prompt*

3.3. Constructing the plant disease map by SLAM

An Agricultural robot is equipped with a camera and RPLidar. Robot will research in greenhouse to collect data of the area to visualized the map of this area. The use of a camera mounted on the robot and the integration of SLAM

(Simultaneous Localization and Mapping) technology into the robot allows farm management to track and store the path the robot has taken. When pests or diseases are detected, the robot will mark the current location on the map created by the robot. At the same time, the robot will also record the direction in which it detected the pests or diseases.

Information about the location and direction of the detected pests or diseases will be transmitted to the farm management department. The management will assign experienced experts to come and handle the detected pests or diseases because some tasks require specialized skills and knowledge that robots cannot perform. This is necessary because pest control may require specialized skills and knowledge that robots cannot perform. Although training robots to perform complex tasks is also a solution, it can be costly and less effective than using human labor.

Integrating SLAM and image processing capabilities into the robot will make autonomous navigation more efficient. The robot can work 24/7 and report any issues with the crops. By combining SLAM, the robot can avoid obstacles such as tools that humans may forget, preventing unnecessary damage.

After the robot recognizes the disease map, it will determine the location of the diseased tree. This is convenient for navigating the robot to the location of the diseased tree to perform tasks such as spraying, removing diseased leaves and some other tasks.

We propose that the next research direction is to combine the plant disease map established in our research with the obstacle map (Fig. 10) to optimize the robot's work in the greenhouse.

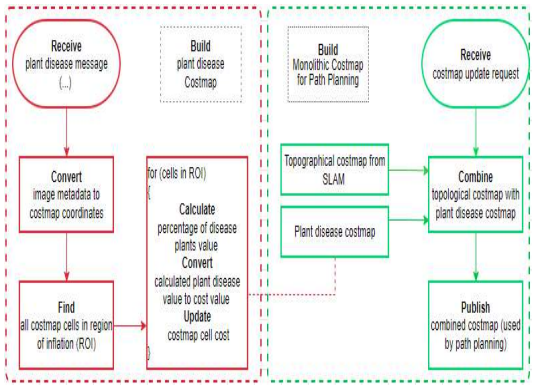

Fig. 10. *Flowchart representing the steps for generating the plant disease costmap and its combination with other costmap layers to provide a monolithic costmap for path planning.*

CONCLUSION

The proposed FastSAM framework demonstrates promising capabilities for enabling autonomous robots to accurately identify multiple plant diseases with minimal training data. By combining the state-of-the-art YOLOv8 object detector and CLIP text encoder, the FastSAM model can be rapidly fine-tuned using just 50-100 annotated examples per disease class. Our experiments show that the fine-tuned FastSAM model achieves approximately 90% test accuracy on real-world plant images captured under uncontrolled field conditions across multiple disease classes. While further validation is

required, these initial results highlight the potential of the FastSAM approach to enable practical autonomous plant disease diagnosis for agriculture robots with low data requirements and minimal training. This will facilitate the deployment of intelligent robotic systems for automated, accurate and scalable crop disease surveillance and care tasks.

The results of the study will be specifically applied in the care of cucumber plants grown in greenhouses at the Vietnam National University of Agriculture.

ACKNOWLEDGMENT

This research was funded by Vietnam's National project "Research, develop an intelligent mobile robot using different types of sensing technology and IoT platform, AI, and implemented in radioactive environment monitoring application", code: DTDLCN.19/23 of the CT1187 Physics development program in the period 2021- 2025.

REFERENCES

[1] Sharma, R., "Artificial Intelligence in Agriculture: A Review", 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021.

[2] Anwar Abdullah Alatawi; Alomani, Shahd Maadi; Najd Ibrahim Alhawiti; Ayaz, Muhammad, "Plant Disease Detection using AI based VGG-16 Model", International Journal of Advanced Computer Science and Applications; West Yorkshire Vol. 13, Iss. 4, 2022.

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al, "Segment anything", 2023.

[4] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, Jinqiao Wang, "Fast Segment Anything", Computer Vision and Pattern Recognition (cs.CV); Artificial Intelligence (cs.AI), 2023.

[5] Juan R.Terven, Diana M.Cordova-Esparza, "A comprehensive review of YOLO: from YOLOV1 and beyond" arXiv:2304.00501 [cs.CV] Computer Vision and Pattern Recognition, Aug 2023.

[6] Armstrong Aboah, Bin Wang, Ulas Bagci, Yaw Adu-Gyamfi, "Real-Time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 5349-5357, 2023.

[7] Mingzhe Hua, Yuheng Lib and Xiaofeng Yang, Empowering Skin Cancer Segmentation with Segment Anything Model. Computer Vision and Pattern Recognition, 2023.

[8] Guorui Xie, Qing Li, Yong Jiang, Tao Dai, Gengbiao Shen, SAM: Self-Attention based Deep Learning Method for Online Traffic Classification Proceedings of the Workshop on Network Meets AI & ML, 2020, pp. 14–20.

[9] Mohsen Ahmadi, Ahmad Gholizadeh Lonbar, Abbas Sharifi, Ali Tarlani Beris, Mohammadsadegh Nouri, Amir Sharifzadeh Javidi, Application of Segment Anything Model for Civil Infrastructure Defect Assessment, Computer Vision and Pattern Recognition (cs.CV), 2023.

[10] Josep Aulinas, Yvan Petillot, Joaquim Salvi, Xavier Lladó, "The SLAM problem: a survey", Ebook, Volume 184: Artificial Intelligence Research and Development, 2008, pp.363 - 371.

[11] Bouchier, P. (2013), "Embedded ROS [ROS Topics]", Robotics & Automation Magazine, IEEE, vol.20, no.2, pp. 17-19, June 2013.

[12] Glenn Jocher, Ayush Chaurasia, and Jing Qiu, "Yolo by ultralytics, https://github.com/ultralytics/ultralytics", 2023.

[13] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al, "Learning transferable visual models from natural language supervision. In International conference on machine learning", 2021, pp. 8748–8763.

[14] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee, "Yolact: Real-time instance segmentation', In Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9157–9166.

[15] Tsung-Yi Lin, Piotr Doll a´r, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie, "Feature pyra- mid networks for object detection", In Proceedings of the IEEE conference on computer vision and pattern recogni- tion, pp. 2117–2125, 2017.

[16] WAS Norzam, HF Hawari, K Kamarudin, "Analysis of mobile robot indoor mapping using GMapping based SLAM with different parameter", IOP Conference Series: Materials Science and Engineering, 2019.

[17] Ngo Quang Uoc, "Studying the improved YOLOv4-tiny algorithms for identifying the powdery mildew and the downy mildew on cucumber" 6th VCCA-2021 conference, 2022.