Object Recognition in Simulation Based on ResNet from Partial Image

Jaehyeon Sung[0009−0009−7692−1168], Kwanho Kim[0009−0004−3094−0746],
Seongmin Kim[0009−0007−8787−0891], Kanghyun Jo[0000−0001−8317−6092]

Dpt. Electrical, Electronic and Computer Engineering, University of Ulsan, Ulsan, South Korea
(jhn6446, aarony12, asdfdsa1234)@mail.ulsan.ac.kr, acejo@ulsan.ac.kr
https://islab.ulsan.ac.kr/

Abstract. This study addresses issues arising from cameras attached to robots. Cameras on robot arms operate in dynamic environments, causing frequent changes in the camera's field of view. As a result, situations where only a portion of an object is identified by the camera are common. To address this, the scenario is limited to a production process with a conveyor belt where a consistent object passes through, and the environment is constructed using Unity to generate a dataset. The model is trained using ResNet to recognize images containing partial objects and estimate the object's position. The significance lies in adjusting the camera's angle based on the model's estimated position to fully capture the object. This research is expected to be beneficial in the increasing use of robotic arms in production processes, addressing problems sequentially in the production process.

Keywords: Robot Arm · Unity · Part of Object · Camera Angle.

1 Introduction

In future industries, robot technology remains crucial, bringing innovative changes not only in industrial sectors but also in diverse fields such as healthcare, transportation, and manufacturing. The core of such transformations lies in the visual recognition capabilities of robots. Particularly, robotic arms heavily rely on cameras to detect and track objects during tasks. Through this, robots comprehend the position, size, and orientation of objects, enabling them to execute precise operations. [3]

However, in specific situations, cameras may observe only parts of objects instead of the entire entity. For instance, a camera attached to a robotic arm may capture only specific sections of an object. In such cases, complete recognition of the object by the robotic arm for accurate task execution can become challenging. Adjusting the position or angle of the camera to observe different parts of the object and comprehend its overall appearance becomes crucial. [4]

This study aims to address issues arising from cameras attached to robots. The scenario assumes a fixed camera position with the freedom to adjust its angle along one axis, restricting the camera's movement to one degree. Additionally, it assumes the object is stationary and has accurate positional information.

The objective of this study is to explore methods for adjusting the camera angle within specified constraints to capture portions of the object. By utilizing the captured images, the goal is to develop a technique for comprehending the overall position of the object. This research plays a crucial role in enhancing the efficiency of robotic operations by improving the effectiveness of camera adjustments or robotic arm tasks in constrained scenarios. [1]

2 Related work

2.1 Unity

Constructed through Unity, the simulation platform utilized in this study, the environment is shaped using the C# programming language for object behavior control. Leveraging the extensive community and ecosystem of Unity, the experimental environment is configured with various resources from the Asset Store, including diverse car types. Additionally, dataset images are acquired by adjusting angles using an internal camera. [5]

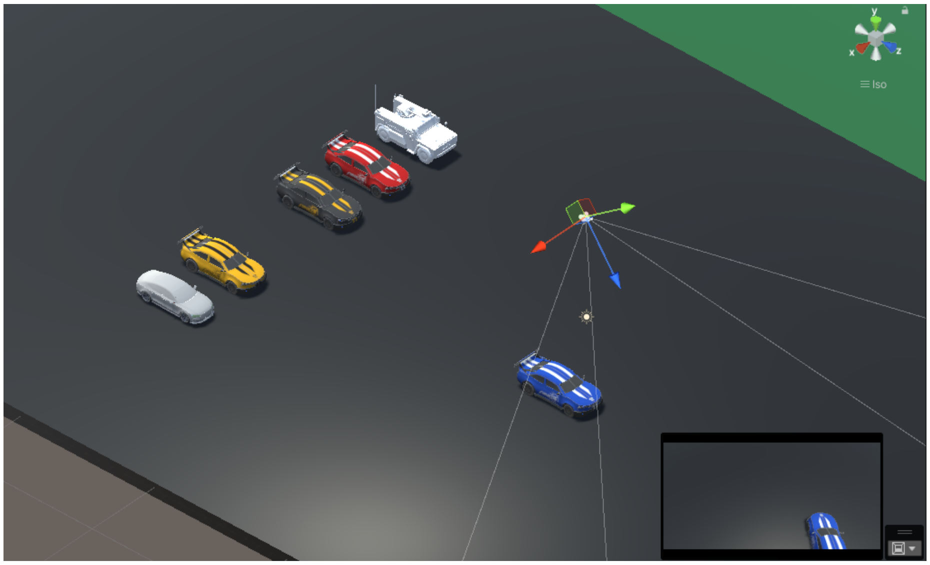

Fig. 1: Simple situation created with Unity, car object imported from assets, and camera position adjustment

2.2 Neural Network

ResNet serves as a fundamental tool in this research. It provides a brief introduction to the deep learning model employed for inferring object positions from images.The model used is ResNet. It is short for Residual Networks, representing an innovative architecture designed for effective deep neural network training. ResNet achieves smooth learning by efficiently propagating gradients through residual connections. In this study, ResNet is applied within the Unity simulation to enhance object recognition in robotic tasks. By identifying features of partially visible objects, the estimation of Z-axis coordinates is performed. To achieve accurate and stable object position estimation, the deep networks are trained using ResNet. The robot's attached camera operates in dynamic environments, leading to frequent changes in the camera's field of view, often observing only parts of objects. Therefore, the efficiency and accuracy of deep neural networks contribute to addressing such scenarios in our research. [2]

3 Proposed method

This paper proposes the following solution to address challenges arising in the process of robotic object recognition for grasping. The dataset is prepared in the Unity simulation environment, featuring images where only parts of objects are visible. Subsequently, training is conducted using ResNet with partially visible images and corresponding label information. The output from the training aims to estimate the values of z_0 and z_1. The scenarios for creating the dataset are as follows:

The camera's position is fixed, and the camera angle is constrained to rotate from 0 to 90 degrees around the X-axis, with the other axes fixed. The object's direction of movement is along the Z-axis. Objects are randomly placed within a specific region captured by the camera, and the capture process proceeds. The partial view of the object captured by the camera, along with the actual coordinates of the object received within Unity (z_0, z_1) is considered. Then, this information is utilized to estimate the angle by which the camera needs to be adjusted to capture the object within its view, denoted as θ_{GT}. The estimation of θ_{GT} is calculated using the following eq.(1)

$$\theta_{GT} = \arctan \frac{2h}{z_0 + z_1} \tag{1}$$

Since the actual positional information of the object is known, the aim is to calculate the camera's angle representing the midpoint. This calculation is intended to position the object within the range of the Z-axis that the camera can capture, based on the Field of View (FoV). Following this, images and label information of cars are input into the proposed ResNet model for training. The model's output provides predicted values for z_0 and z_1, enabling the estimation of the object's position based on the input image.

The training proceeds with the dataset constructed through this process, consisting of images and corresponding object coordinates. The training model

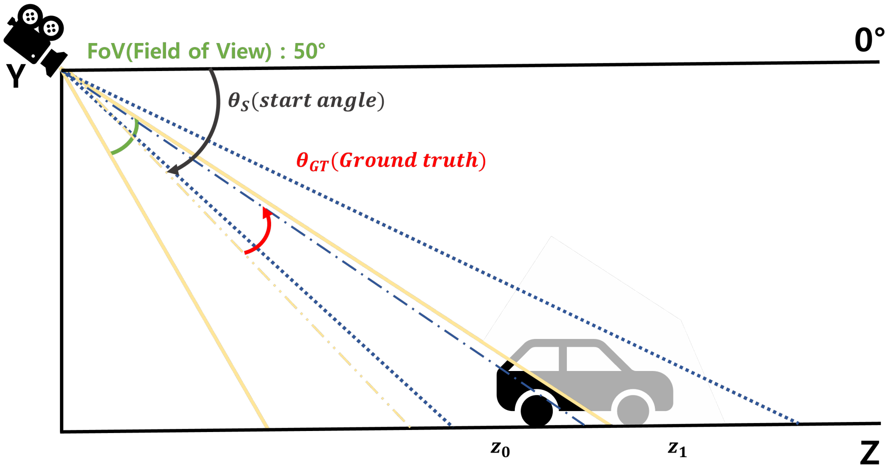

Fig. 2: Visualization of the angle at which the camera is adjusted and positioning where only part of the car is visible

utilizes ResNet, as explained earlier. During the training, the loss function used is defined by the following equation eq.(2)

$$\mathcal{L} = \frac{1}{n} \sum_{i} \left(\left(z_0^i - \hat{z}_0^i\right)^2 + \left(z_1^i - \hat{z}_1^i\right)^2 + \left(\theta_{GT}^i - \hat{\theta}_{GT}^i\right)^2 \right) \tag{2}$$

4 Experiment

All scenarios are conducted within the Unity simulation environment. The creation of products and their passage on a conveyor belt is implemented and simulated in a simplified manner.

4.1 Dataset

The dataset was directly constructed within the simulation. Objects were randomly placed within a specific area, and the dataset was built by adjusting the camera angle. With each image capture, the z_0, z_1, and θ_{GT} values of the object were saved as labels in a txt file. The camera angles ranged from 0° to 90°, increasing by 5° increments, resulting in a total of 5,700 images, with 300 images for each angle. Images were randomly arranged, leading to a substantial number of images where the camera did not capture the object. However, it is expected that if the robot arm does not capture the image, the robot's movement should increase. Therefore, this data was also applied in both training and testing.

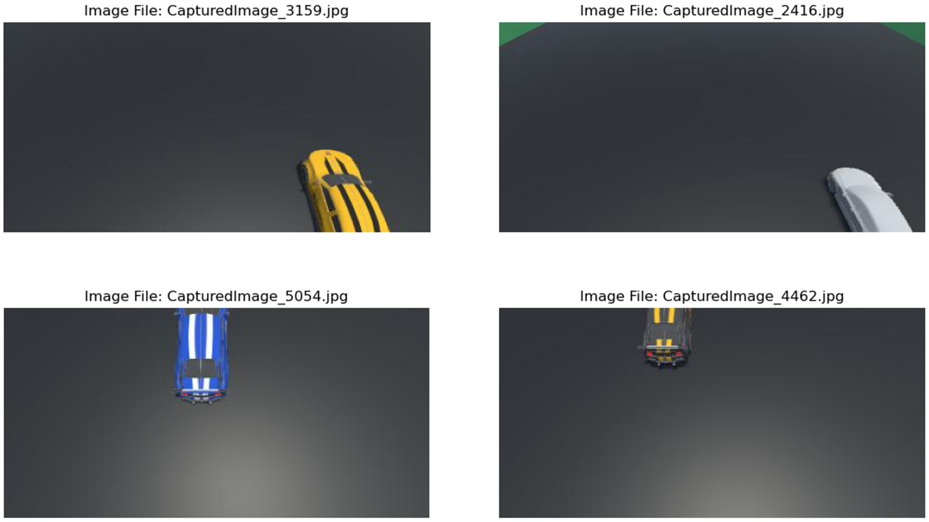
Image File: CapturedImage_3159.jpg
Image File: CapturedImage_2416.jpg
Image File: CapturedImage_5054.jpg
Image File: CapturedImage_4462.jpg

Fig. 3: Dataset sample in the Unity simulation environment featuring images where only parts of objects are visible

4.2 Evaluation metrics

Utilizing Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE) as evaluation metrics provides a comprehensive analysis. MAE represents the average absolute difference between predicted and actual values, where lower values signify higher model accuracy. MSE computes the average of squared differences, considering the magnitude of errors in predictions. Lastly, MAPE, with values between 0 and 100, offers an easily interpretable probability scale. Through these metrics, the aim is to assess object estimation performance from various perspectives and reliably analyze the experimental results.

4.3 Implementation setup

The experiments will be conducted in the following environment: Intel Core i9-9960X, NVIDIA 2080 Ti x 4EA, and 125.5GB of memory. The training process for the experiments will span 500 epochs, with a batch size of 64, a learning rate of 0.001, and the Adam optimizer.

5 Conclusion

This study proposes a cognitive approach utilizing deep learning for the detection of partially visible objects through robotic arm vision. Using Unity simulation, scenarios were simulated where objects pass through a conveyor belt, capturing a dataset with only partial views in the camera. The dataset was constructed with angles ranging from 0° to 90° in 5° increments, totaling 5,700 images with 300 images for each starting angle.

The plan involves preprocessing the dataset by removing irrelevant information and structuring the data for training in the future. Additionally, there are intentions to increase the number of epochs for enhanced predictive performance. If the improvement proves insufficient, consideration will be given to reducing input labels to z_0 and z_1 for estimating only the object's position. The structure will also be modified to compute θ_{GT} through operations. Furthermore, in this study, the camera's degree of freedom is limited to 1. However, currently commercialized robotic arms typically have at least 6 degrees of freedom. The future goal is to increase the camera's freedom for more accurate object localization and incorporate SLAM technology to explore the surrounding environment in cases where the object is not captured by the camera.

This research is anticipated to be beneficial in the increasing use of robotic arms in production processes in the future. When producing products with robotic arms, there are likely to be many environmental constraints and unexpected events. This study aims to reduce such events to enhance the efficiency and reliability of production processes involving robotic arms.

6 Acknowledgements

This results was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003)

References

1. Besl, P.J., Jain, R.C.: Three-dimensional object recognition. ACM Comput. Surv. **17**(1), 75–145 (mar 1985). https://doi.org/10.1145/4078.4081, https://doi.org/10.1145/4078.4081
2. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
3. Hesselroth, T., Sarkar, K., van der Smagt, P., Schulten, K.: Neural network control of a pneumatic robot arm. IEEE Transactions on Systems, Man, and Cybernetics **24**(1), 28–38 (1994). https://doi.org/10.1109/21.259683
4. Kim, K., Kim, J., Jo, K.: Top-down pose estimation method based human-computer interaction for smart space system with digital twin. In: 2023 International Conference on Intelligent Metaverse Technologies & Applications (iMETA). pp. 1–5. IEEE (2023)
5. Yang, C.W., Lee, T.H., Huang, C.L., Hsu, K.S.: Unity 3d production and environmental perception vehicle simulation platform. In: 2016 International Conference on Advanced Materials for Science and Engineering (ICAMSE). pp. 452–455 (2016). https://doi.org/10.1109/ICAMSE.2016.7840349