Educational Administration: Theory and Practice
2024, 30(4), 7341-7346
ISSN: 2148-2403
https://kuey.net/

Research Article

YOLO-Based Real-Time Monitoring System Using Radar And Cameras

Seung-Yeon Hwang[1*], Jeong-Joon Kim*

[1*]Dept. of Computer Engineering, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea, E-mail: syhwang@gs.anyang.ac.kr

*Dept. of Software, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea, E-mail: jjkim@anyang.ac.kr

Corresponding Author: Jeong-Joon Kim, Dept. of Software, Anyang University, Anyang-si, Gyeonggi-do, Republic of Korea, E-mail: jjkim@anyang.ac.kr

Citation: Seung-Yeon Hwang et al, (2023), YOLO-Based Real-Time Monitoring System Using Radar And Cameras, *Educational Administration: Theory And Practice, 3(4), 7341-7346*
Doi: 10.53555/kuey.v30i4.2567

ARTICLE INFO

ABSTRACT

Recently, object recognition technology is positioned as a core technology in the overall shipbuilding industry, such as video surveillance, face recognition, robot control, autonomous driving, smart factory, and security, due to the advancement of hardware performance and miniaturization and related technologies. In this paper, we intend to develop a deep learning-based real-time monitoring system using data collected from radar and cameras. The camera is moved to the corresponding location using the location information of the object detected by the radar. In addition, the AI analysis client recognizes objects present in the image at the corresponding location and transmits the result to the server. Finally, the server determines whether there is a threat according to the object information received from the AI analysis module and sounds a warning sound. In order to develop such a system, first, an AI analysis module development environment for object recognition in an image is established, and a protocol necessary to communicate data between a server and an AI analysis module is defined. And the method to improve the real-time and accuracy of the AI analysis module applies.

Keywords: Object Detection, Deep Learning, real-time monitoring, AI, Camera, Radar

1. Introduction

With the recent development of science and technology, the demand for artificial intelligence and big data technologies is increasing, and research to integrate these technologies in various fields is being actively conducted. Recently, due to the high performance and miniaturization of hardware, there has been an increasing attempt to introduce deep learning based real-time object recognition technology in embedded devices. Accordingly, several manufacturers are developing and launching small devices dedicated to deep learning in line with these needs. However, there are limitations in direct learning and inference due to the lack of performance of devices such as lightweight devices, mobile devices, and IoT sensors. Therefore, research on reducing the weight of deep learning models is being actively conducted to simplify operations while maintaining the accuracy of the model level learned in the existing high-performance server [1].

Past object recognition studies have been a method of finding objects by detecting features of objects such as Scale Invariant Feature Transform (SIFT) [2], Speed-Up Robust Features (SURF) [3], Haar [4], and Histogram of Oriented Gradients (HOG) [5]. However, with the recent emergence of CNN (Convolutional Neural Network), which won the ImageNet competition, deep learning-based object recognition methods have become mainstream [6]. The problem of recognizing and classifying objects in an image achieved excellent performance, but there was a limit to detecting the position of objects in the image. Therefore, research has emerged to compensate for these limitations.

Region-based Convolutional Neural Networks (R-CNN) [7] is an initial study that solved the problem of object location detection. However, due to the disadvantage of slow speed, it could not actually be used in object recognition and detection systems, and several methodologies that supplemented this have emerged. The emerging technologies significantly improved object recognition and detection speed but were not sufficient to apply them to areas such as surveillance systems, robots, and autonomous driving that required processing speeds close to real-time. You Only Look Once (YOLO) [8] proposed a method of configuring all processes of object recognition into one deep learning network to solve this speed problem.

In this paper, a system for real-time monitoring is developed by applying deep learning-based object recognition technology to data collected from radar and cameras. The radar first detects an object and moves the camera to the corresponding position using the location information of the detected object. The AI analysis module transmits information obtained by recognizing and classifying objects to the server using an image of a corresponding location collected from the camera. The server determines whether there is an illegal intrusion according to the object information and sounds a warning sound.

2. Related Works and Dataset

This chapter describes the techniques for recognizing and classifying objects from images collected from the camera and the dataset used to train the deep learning model.

2.1 YOLO

YOLO simplified the object detection process to a single regression problem, eliminating the need for a separate network for extracting areas of interest and significantly improving the training and detection speed of the model. Unlike conventional sliding windows or region proposal methods, YOLO searches the entire image at the training and test stage to learn and process not only information on the shape of the class but also surrounding information. In particular, YOLO is worth sufficient utilization in various services because it has higher detection accuracy for new images that have not been trained by learning up to the general part of the object. As a result of experiments with the PASCAL VOC 2007 and 2012 datasets, the YOLO researchers showed significantly faster processing speed than the conventional method by processing 45 frames per second for YOLO and 155 frames per second for Fast YOLO. However, there was a disadvantage that the recognition accuracy was somewhat inferior. Figure 1 shows the object recognition result using YOLO.

Fig. 1 Object recognition using YOLO

2.2 Dataset

In this paper, the Exclusively Dark (ExDARK) [9] dataset and the Common Objects in Context (COCO) [10] dataset were used for model training to improve the object recognition accuracy of the AI analysis module.
In general, a monitoring system is used as a means for filling boundary gaps that may occur at night rather than during the day when visual identification and monitoring are easy. Therefore, the ExDARK dataset is used to enhance the accuracy of detecting and recognizing new objects at night, that is, in a low-light environment. The ExDARK dataset consists of 7363 low-light images in the evening from a very dark environment. A total of 12 types of objects are annotated in all images. (The ExDARK dataset is a collection of 7,363 low-light images from very low-light environments to twilight with 12 object classes annotated on both image class level and local object bounding boxes.) Figure 2 shows a sample of the ExDARK dataset.

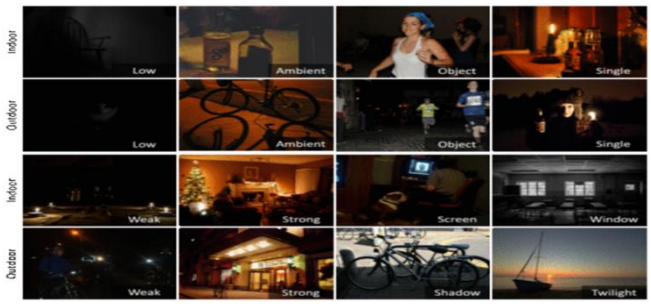

Fig. 2 ExDARK sample dataset

The COCO dataset is a dataset created for the purpose of (computer vision) tasks such as (object detection), (segmentation), and (keypoint detection), and is one of the datasets widely used for performance evaluation in various studies. The COCO dataset consists of 118,000 training images, 5000 validation images, and 41000 test images for a total of 80 types of objects. Figure 3 shows a sample of the COCO dataset.

Fig. 3 COCO sample dataset

3 Research contents and results

3.1 System architecture and operation method

In the monitoring system to be developed in this study, one radar and two cameras (real image camera, thermal image camera) are used. In a low-light environment such as at night, it is difficult to recognize a detected object due to a problem such as a deterioration in image quality of a real image camera. Therefore, a thermal imaging camera was used together to solve this problem and minimize the monitoring gap. In order to communicate information on detected and recognized objects, one server and one AI analyzer are used, and in the AI analyzer, a real-image AI analysis client and a thermal-image AI analysis client access one server to exchange data. Figure 4 shows the overall architecture of the monitoring system.

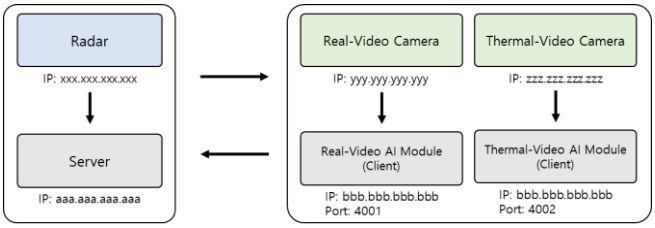

Fig. 4 Real-time monitoring system architecture

The server waits for requests from the real image AI analysis client connected to port 4001 and the thermal image AI analysis client connected to port 4002. After the connection between the server and the client is completed, the server receives the location information of the detected object from the radar, and moves the real image camera and the thermal image camera to the location where the object exists by using the information. When the movement command is executed, the server transmits an analysis request for the object to the real image and thermal image AI analysis module. When each AI analysis module receives a request from the server, it uses the image frame after waiting for 3 seconds for camera shake and focus correction for object recognition. When object recognition is completed, information about the object is sent to the server and the server waits for an analysis request. The server determines whether there is a threat through the object information received from the AI analyzer, and sounds a warning sound if it corresponds to a threat such as illegal intrusion.

In addition, research was conducted to improve the reliability and secure real-time properties of object recognition results derived by real-image and thermal-image AI analysis modules. In the initial system, the Tiny model was used to reduce the hardware load of each AI analysis module. And, it always analyzes the video frame, waits for the server's request, and sends the analysis result for one frame immediately after the request is received. Although this method secured real-time performance, there was a disadvantage in that the accuracy was very low. Therefore, instead of analyzing only one frame, the object recognition results for 5 frames were synthesized and transmitted to the server to improve accuracy. As a result, the accuracy was improved compared to the first method, but it was judged that it was insufficient for use in the surveillance system due to problems such as the limitation of the Tiny model and the image delay of the AI analysis module. Therefore, the accuracy was improved by applying a model with excellent performance instead of the Tiny model. It solves the video delay problem and hardware load, and waits until the server's analysis request occurs instead of constantly analyzing for real-time. When an analysis request occurs, object recognition is performed on 5 frames, and the results are aggregated and transmitted to the server. In this way, high accuracy and real-time were achieved.

3.2 Development Environment

For the server, it uses Windows 10 Pro OS and is equipped with an AMD Ryzen 7 3700X 8-Core Processor and 16 GB of memory. The AI analyzer used NVIDIA Jetson AGX Xavier, and the object recognition program was implemented using Python and Tensorflow.

3.3 Communication Protocol Definition

3.3.1 Frame structure

A transmitted message frame is composed of a header part having a fixed length, a data part having a variable length according to a command code in the header, and an end part having a fixed length. Also, byte ordering (Network Ordering: Little-Endian) is followed across all packets. Table 1 indicates the description of the type of field type.

Tab. 1 Field Type Type

type code	*Explanation*
N	Integer (1, 2, or 4 bytes long)
C	It is used for a field with a string and does not contain NULL. However, when setting information shorter than the field length, the blank part may be set to NULL.
B	It is used for fields that are handled in units of bytes or bits.

In the case of field length, it is generally expressed as an integer number, and when expressing a variable length, it is expressed as [N] (a field having a length of N bytes). Table 2 indicates the message frame structure, and descriptions of each field are written in Table 3.

Tab. 2 Message Frame Structure

field name	*STX*	*OP*	*Length*	*DATA*	*ETX*
field type	B	B	N	-	B
field length	1	1	2	[N]	1
	header part			data part	end part

Tab. 3 message frame field

field name	*Explanation*
STX	□ field value : 0x02 (1Byte) □ abbreviation : Start of Text □ definition : Delimiter to separate the start of transmitted data
OP	□ field value : BIN (1Byte) □ abbreviation : packet Operand □ definition : Indicate the command of the transmit/receive frame
Length	□ field value : BIN (2Byte) □ abbreviation : data Length (High and Law byte) □ definition : length of DATA part
DATA	□ field value: For field value definition, refer to data definition and analysis ([N]Byte) □ abbreviation : DATA □ definition : field equipment and center creation data
ETX	□ field value : 0x03 (1Byte) □ abbreviation : End of Text □ definition : a separator for separating the end of the transmission data

3.3.2 Object identification information request and response

When the radar detects an object, the server sends a PTZ command. The server requests object identification information from the AI analysis module after the PTZ movement time and image stabilization time. The server sends the detection time of the object, radar number, camera number, and AI analyzer number to the AI analysis module, and the AI analysis module sends the object recognition result in the image to the server. Table 4 indicates information about a message requesting object identification information from the server to the AI analysis module, and Table 5 indicates response message information for object identification information from the AI analysis module.

Tab. 4 Server -> AI analysis module (OP: 0xA0) object identification information request message

Length		***10 BYTE***		
No	*field name*	*field type*	*field length*	*field description*
1	ControllerID	N	1	- controller unique number (ex:20)
2	RadarID	N	1	- radar unique number (ex:25)
3	AiClassifierID	N	1	- target AI discriminator unique number (ex:26)
4	DetectDateTime	N	7	- Time of occurrence of object information - year(2Byte), month(1Byte), day(1Byte), hour (1Byte), minute(1Byte), second(1Byte)

Tab. 5 AI Analysis Module -> Server (OP: 0xA0) Object Identification Information Response Message

<table>
<tr><td colspan="2">Length</td><td colspan="3">12+10*ObjectCount BYTE</td></tr>
<tr><td>No</td><td>field name</td><td>field type</td><td>field length</td><td>field description</td></tr>
<tr><td>1</td><td>ControllerID</td><td>N</td><td>1</td><td>- Controller identification number (ex:20)</td></tr>
<tr><td>2</td><td>RadarID</td><td>N</td><td>1</td><td>- Radar identification number (ex:25)</td></tr>
<tr><td>3</td><td>CameraID</td><td>N</td><td>1</td><td>- camera identification number (ex:21)</td></tr>
<tr><td>4</td><td>AiClassifierID</td><td>N</td><td>1</td><td>- AI identifier unique number (ex:26)</td></tr>
<tr><td>5</td><td>ClassifiedDateTime</td><td>N</td><td>7</td><td>- Time of occurrence of object identification information (time immediately before response)
- year(2Byte), month(1Byte), day(1Byte), hour (1Byte), minute (1Byte), second (1Byte)</td></tr>
<tr><td>6</td><td>ObjectCount</td><td>N</td><td>1</td><td>- number of identified objects (0..255)</td></tr>
<tr><td colspan="2">Length</td><td colspan="3">10 BYTE * ObjectCount repeat</td></tr>
<tr><td>7</td><td>ObjectClassType</td><td>N</td><td>1</td><td>- class type range: 00~79, unknown=100
<table>
<tr><td>No.</td><td>class number</td><td>class name</td></tr>
<tr><td>1</td><td>00</td><td>person</td></tr>
<tr><td>2</td><td>01</td><td>bicycle</td></tr>
<tr><td>3</td><td>02</td><td>car</td></tr>
<tr><td>4</td><td>03</td><td>motorbike</td></tr>
<tr><td>5</td><td>05</td><td>bus</td></tr>
<tr><td>6</td><td>07</td><td>truck</td></tr>
<tr><td>7</td><td>14</td><td>bird</td></tr>
<tr><td>8</td><td>15</td><td>cat</td></tr>
<tr><td>9</td><td>16</td><td>dog</td></tr>
<tr><td>10</td><td>19</td><td>cow</td></tr>
<tr><td>11</td><td>21</td><td>bear</td></tr>
</table></td></tr>
<tr><td>8</td><td colspan="2">AccuracyPercent</td><td>N</td><td>1</td><td>- accuracy %(0..100)</td></tr>
<tr><td rowspan="2">9</td><td rowspan="2">LTpoint</td><td>x</td><td>N</td><td>2</td><td>- upper left (LeftTop)</td></tr>
<tr><td>y</td><td>N</td><td>2</td><td>- upper left (LeftTop)</td></tr>
<tr><td rowspan="2">10</td><td rowspan="2">RBpoint</td><td>x</td><td>N</td><td>2</td><td>- bottom right (RightBottom)</td></tr>
<tr><td>y</td><td>N</td><td>2</td><td>- bottom right (RightBottom)</td></tr>
<tr><td colspan="2">……</td><td colspan="2">….</td><td>….</td><td>…..</td></tr>
</table>

4 Conclusion

The YOLO-based real-time monitoring system using radar and camera designed and developed in this paper uses one radar, real image and thermal image camera and uses one server and AI analyzer to communicate information about detected and recognized objects. In the AI analyzer, the real image AI analysis client and the thermal image AI analysis client connect to one server and data is exchanged according to the defined communication protocol. In addition, the present invention can improve the reliability and real-time of the monitoring system by performing object recognition for five frames only when a request is received from a server without using a tiny model in a real image and a thermal image AI analysis module and transmitting the results to a server after synthesizing the analysis results. However, the monitoring system developed in this paper cannot guarantee 100% accuracy of all detected objects, so it is expected that it will be used as a monitoring and boundary system for primary response to potential threats. In the future, we will study the object recognition technology based on the tensorRT optimized for embedded devices and expect to mitigate the hardware load of Nvidia Jetson AGX Xavier used in this study and achieve dramatic performance improvement.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1F1A1062953).

References

1. Y.J. Lee, Y.H. Moon, J.Y. Park, O.G. Min, Recent R&D Trends for Lightweight Deep Learning, ETRI, 2019.
2. D.G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” Int. J. Comput. Vision, vol. 60,

no. 2, 2004, pp. 91-110.
3. H. Bay et al., “Speeded-Up Robust Features (SURF),” Comput. Vision Image Understanding, vol. 110, no. 3, 2008, pp. 346-359.
4. P. Viola and M. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” Proc. IEEE Conput. Soc. Conf. Comput. Vision Pattern Recogn., Kauai, HI, USA, Dec. 8-14, 2001, pp. I:511-I:518
5. N. Dalal and B. Triggs, “Histograms of Oriented Gradients for Human Detection,” IEEE Comput. Soc. Conf. Comput. Vision Pattern Recogn., San Diego, CA, USA, June 20-25, 2015, pp. 886-893.
6. A. Krizhevsky, I. Sutskever and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105, 2012.
7. R. Girshick et al., “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation,” IEEE Conf. Comput. Vision Pattern Recogn., Columbus, OH, USA, June 23-28, 2014, pp. 580-587.
8. J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” IEEE, Conf. Comput. Vision Pattern Recogn., Las Vegas, NV, USA, June 27-30, pp.779-788.
9. Loh, Y. P., & Chan, C. S. (2019). Getting to know low-light images with the exclusively dark dataset. Computer Vision and Image Understanding, 178, 30-42.
10. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham.