International Journal of Advanced Culture Technology Vol.10 No.2 300-306 (2022)
DOI https://doi.org/10.17703/IJACT.2022.10.2.300

IJACT 22-6-39

Detecting Jaywalking Using the YOLOv5 Model

[1]Hyun-Tae Kim, [2]Sang-Hyun Lee

[1]Student, Department of Computer Engineering, Honam University, Korea
[2]Associate Professor, Department of Computer Engineering, Honam University, Korea
[1]gusxo0560@naver.com, [2]leesang64@honam.ac.kr

Abstract

Currently, Korea is building traffic infrastructure using Intelligent Transport Systems (ITS), but the pedestrian traffic accident rate is very high. The purpose of this paper is to prevent the risk of traffic accidents by jaywalking pedestrians. The development of this study aims to detect pedestrians who trespass using the public data set provided by the Artificial Intelligence Hub (AIHub). The data set uses training data: 673,150 pieces and validation data: 131,385 pieces, and the types include snow, rain, fog, etc., and there is a total of 7 types including passenger cars, small buses, large buses, trucks, large trailers, motorcycles, and pedestrians. has a class format of Learning is carried out using YOLOv5 as an implementation model, and as an object detection and edge detection method of an input image, a canny edge model is applied to classify and visualize human objects within the detected road boundary range. In this study, it was designed and implemented to detect pedestrians using the deep learning-based YOLOv5 model. As the final result, the mAP 0.5 showed a real-time detection rate of 61% and 114.9 fps at 338 epochs using the YOLOv5 model.

Keywords: *YOLOv5 Model, Object Detection, Canny Edge, Hough Transform, Trespassing*

1. INTRODUCTION

According to the Traffic Accident Analysis System (TAAS) traffic data for 2021 of the Road Traffic Authority, out of 210,000 traffic accidents in 2020, the number of deaths is 3,081 and the number of injuries is about 300,000. In Table 1, trespassing traffic accidents accounted for 2.9% of 210,000 cases, and the fatality rate was 10.9% out of 3,081 cases, indicating that trailing traffic accidents have a high mortality rate [1-2].

Table 1. Ratio of jaywalking accidents

classification	Sub-category	2016year	2017 year	2018 year	2019 year	2020 year
Jaywalking accident	Number of accidents	6.695%	4.433%	4.127%	3.938%	2.969%
	Number of deaths	16.519%	13.429%	13.700%	13.616%	10.938%
	Number of injured	4.349%	2.866%	2.679%	2.567%	1.972%

source: TAAS (2021)

According to statistics like this, pedestrian traffic accidents can be viewed as high-risk accidents that lead to the most deaths. Pedestrian traffic accidents are often caused by momentary errors of judgment or mistakes,

Manuscript received: May 28, 2022 / revised: June 5, 2022 / accepted: June 9, 2022
Corresponding Author: leesnag64@honam.ac.kr
Tel: +82-62-940-5285, Fax: +82-62-940-5285
Associate Professor, Department of Computer Engineering, Honam University, Korea

even if the driver of a vehicle drives with caution. Alternatively, it often occurs when a pedestrian does not recognize that a vehicle is coming, or an elderly person or a child does not cross within a valid signal. But even more dangerous is the act of pedestrians ignoring the signal or crossing the road without using the crosswalk [3].

Although this risk is high, the infrastructure system for preventing pedestrian accidents in Korea is insufficient. Currently, in the case of Intelligent Transport Systems (ITS), Korea's representative intelligent transportation system, traffic information is collected based on the Global Positioning System (GPS) in the vehicle for smooth traffic flow in and outside the city center based on Vehicle to Infrastructure (V2I). It is not possible to intelligently detect dangerous situations and deliver real-time information because its main purpose is to distribute and distribute information. In particular, it is difficult to respond to a dangerous situation for a corner escape section when it is raining or snowing, or when visibility is not secured due to bad weather conditions such as fog, or when a long-curved corner is passed.

In this paper, to prevent the problem of high risk of traffic accidents for jaywalking pedestrians, we try to detect pedestrians who jaywalk using the data set provided by Artificial Intelligence Hub (AIHub). The data set uses training data: 673,150 pieces and validation data: 131,385 pieces. The types of data include snow, rain, and fog, and there is a total of 7 class types: passenger cars, small buses, large buses, trucks, large trailers, motorcycles, and pedestrians.

As the implementation model of this study, learning is carried out using the YOLOv5 model, learning and image edges are detected using the canny edge method and object detection of the input image, and objects within the detected road boundary are classified and visualized.

2. RESEARCH METHOD

2.1 YOLOv5 Model

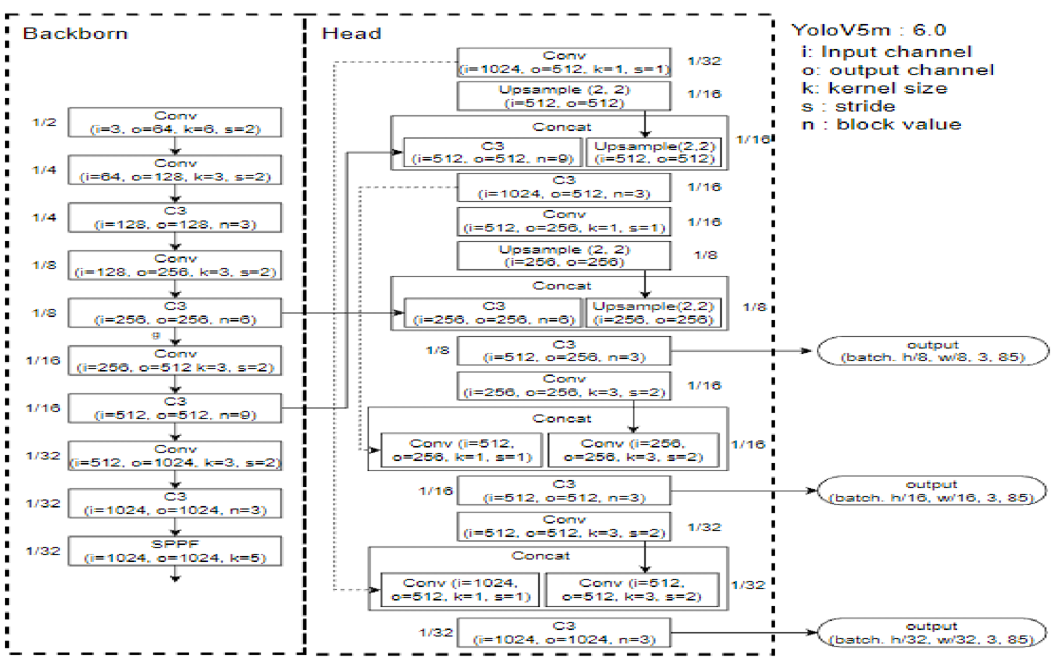


Figure 1. YOLOv5 model Architecture

This paper selects the YOLOv5 model of deep learning and uses it to detect pedestrians and road conditions. The You Only Look Once (YOLO) model with a one stage detector structure has versions such as YOLO,

YOLO:9000, YOLOv3, and YOLOv4 apply [4-6]. Here, the model of YOLOv5 consists of a total of 5 versions: Nano, Small, Medium, Large, and XLarge, and in this study, the medium size is used for learning.

Figure 1 shows the Backbone and Head of the YOLOv5 Medium model. In the Backbone, the feature map is extracted from the image, and the image size has a structure that extracts and detects branching results in three different sizes [7].

2.2 Canny Edge Algorithm

Canny edge algorithm is the most used algorithm in edge detection method. In general, edge detector is very sensitive to noise. Therefore, it is an algorithm developed to avoid calculating false boundaries due to noise. The Canny algorithm goes through the following five steps. First, we remove the noise from the image with a Gaussian filter. Second, the intensity of the gradient is obtained using the Sobel filter. Third, non-maximum suppression is applied to remove false responses from the boundary line detector. Fourth, a double threshold method is applied to select probable pixels as boundaries. Fifth, in the previous double threshold method, the part that exceeds maxVal is set as a strong edge, and the part between minVal and maxVal is set as a weak edge, and the weak edge connected to the strong edge is judged as an edge, and the part that does not is removed [8].

3. IMPLEMENTATION

Figure 2 shows the development process used in this paper and the details are as follows.

The first step is to secure data. Among public data sets including pedestrian crossings, people, and cars, the dataset provided by AIHub is used [9]. The dataset includes snow, rain, and fog, and includes a total of seven classes: passenger cars, small buses, large buses, trucks, large trailers, motorcycles, and pedestrians.

Second, the training weights of YOLOv5 are selected and the model is trained using the processed data set. The third stage is a visualization part that detects, detects, and visualizes objects within the boundary range of the road shaped by object detection and edge detection of the input image.

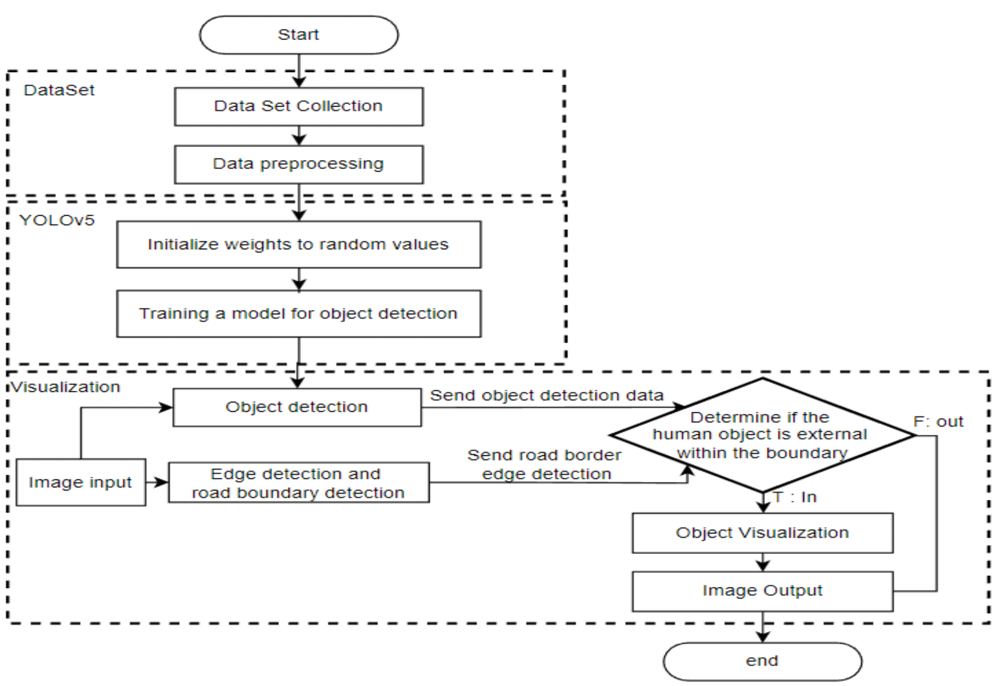


Figure 2. Structure of the development process

3.1 DataSet

For the configuration of the data set, a public data set provided by AIHub including pedestrians, people, and cars is used [9]. The types of data used include snow, rain, and fog, and are divided into passenger cars, small buses, large buses, trucks, large trailers, motorcycles, and pedestrians, and there is a total of 7 classes.

This study proceeds with training with total training data: 673,150 pieces and validation data: 131,385 pieces.

Table 2. Training DataSet

Data organization	Training data	Validation data
Car	500,358	97,593
Small bus	1,785	458
Large bus	22,902	4,441
Truck	76,232	13,905
Large trailer	1,014	27
Motorcycles or bicycles	9,891	2,486
Pedestrian	60,968	12,475
Total value	673,150	131,385

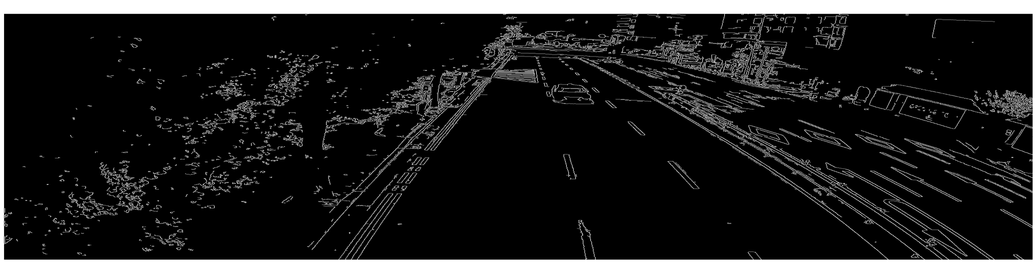
3.2 Edge Detection

The first line of Table 3 is a code that implements the canny edge and shows the execution result.

In the case of canny edge detection, if the lower threshold is set low, the correct edge result is not shown, and if the upper threshold is set high, the edge is treated too strongly. Therefore, in this paper, the low threshold value is set to 50, and a canny edge set to a high threshold value of 200 is applied for visualization [10].

Table 3. Canny Edge Code & Result

No.	Code & Result
1	canny = cv2.Canny(img, 50, 200, None, 3)



4. RESULT

The development environment of this paper was implemented in a hardware environment of Ubuntu 20.04.4 LTS operating system, CPU performance of Intel i9-10900K, GPU performance of NVIDIA RTX 6000 24GB, RAM 128GB, and software environment of Python 3.7.13, CUDA 11.5.119 Development is carried out in the environment of cuDNN 8.4.0, PyTorch 1.11.0, Torchvision 0.12.0, and openCV 4.5.5.64.

In YOLOv5 training, batch-size was set to 32, image size was set to 640x640, and the epoch value was set to 400.

Table 4 shows the Training Loss values of YOLOv5. Comparing the content of training 1 steps and training 200 steps, the value change shows about 0.034 for object loss (obj loss), about 0.051 for bounding box loss(Box loss), and about 0.0026 for classification loss (cls loss). In the case of 300 steps to 320 steps with increased number of training, the value change shows 0.002 for obj loss, 0.0035 for Box loss, and 0.00013 for cls loss. Therefore, it was judged that further measurement was unnecessary and the learning was terminated early at 338.

Table 4. Training Loss

Epoch	Obj loss	Box loss	Cls loss
1	0.08034	0.08896	0.03313
100	0.04763	0.03941	0.00759
200	0.04635	0.03825	0.00694
300	0.04540	0.03748	0.00635
320	0.04520	0.03713	0.00622
338	0.04467	0.03691	0.00616

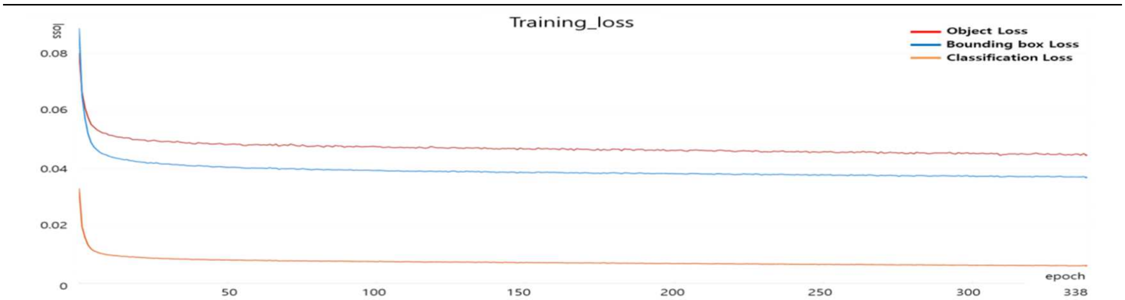


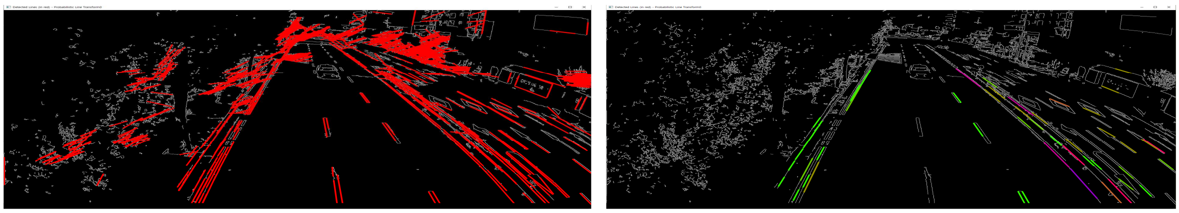
Table 5 shows the metrics of the learning process of YOLOv5. It can be seen that the fluctuation range of precision and recall is large before learning 250 times, but it can be seen that the fluctuation range is gradually reduced thereafter. Also, in YOLOv5, the value of mean Average Precision (mAP) 0.5, which means accuracy, was 58.9% when learning 100 epochs, and the value gradually increased according to the number of training sessions. As the final model, a model trained 338 times that reached an accuracy of 60.5% was selected as the final model.

Table 5. Training metrics

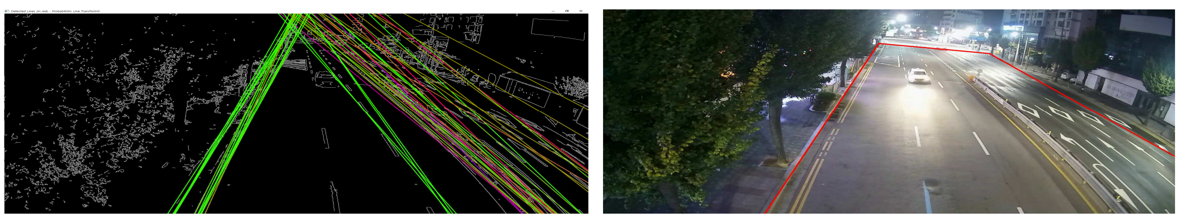
Epoch	Precision	Recall	mAP 0.5	mAP 0.5:0.95
1	0.5600	0.1515	0.0987	0.0457
100	0.6885	0.5759	0.5889	0.4157
200	0.7079	0.5890	0.5945	0.4198
300	0.7167	0.5701	0.5932	0.4204
320	0.6896	0.5900	0.5931	0.4206
338	0.6766	0.5976	0.6053	0.4236

4.1 Edge Detection Applied Result

Figure 3 visualizes edge detection step by step. Figure 3(a) is shows the step of extracting the boundary between the road and the sidewalk by detecting the canny edge and applying the Hough transform. The Figure 3(b) shows the result of preprocessing without marking the road boundary due to the detection result of street trees or buildings and cars in the Hough transform step. In the preprocessing process, the horizontal line is removed because the road in the straight lane is directed in the vertical direction. The Figure 3(c) shows the result of visualization of the length of the preprocessed line using cos and sin values based on the length of the y-axis. The Figure 3(d) shows the visualization result using the values of the minimum and maximum distances on the x-axis [11].



(a) Edge detection and Hough transform original (b) Pretreatment



(c) Edge processing boundary line visualization (d) Road boundary line visualization result

Figure 3. Visualization results

The values of Non-Maximum Suppression Intersection of Union threshold (NMS IoU threshold) and confidence threshold applied in this paper are set to 0.6. The reason for this setting is that if the value of the NMS IoU threshold is set lower than 0.6, the size of the object box appears larger than that of the object. In addition, if the confidence threshold is set to a low value, an error occurs in the detection result as the reliability of the object is lowered.

①YOLOv5 Test Result ②trespassing detection

FPS	Precision	Recall	mAP 0.5
114.9	68%	60%	61%

Figure 4. YOLOv5 Result

① in Figure 4 shows the result screen without edge detection applied, and ② on the right is the value of detecting trespassing by applying edge detection.

In this paper, the test result of the YOLOv5 model reached 114.9 frames per second (FPS), and the precision was 68%, recall 60%, and mAP 61%. Pedestrians trespassing were detected according to the detection position of the pedestrian.

5. CONCLUSION

Currently, jaywalking, in which pedestrians cross the street without ignoring signals or using crosswalks, is a dangerous act that accounts for 10% of pedestrian traffic fatalities. The infrastructure system to prevent and prevent such pedestrian accidents is insufficient. In the case of ITS, Korea's representative intelligent transportation system, it is based on V2I. Because the main purpose is to collect and distribute traffic information, it is not possible to intelligently detect dangerous situations and deliver real-time information.

Therefore, in this paper, to solve the problem of high mortality rate for jaywalkers, using data provided by AIHub, YOLOv5 model and edge detection were applied to develop jaywalking detection. As a result of the test of the YOLOv5 model used in the development, the detection speed of jaywalking reached 114.9 FPS, the precision was 68%, the recall was 60%, and the mAP was 61%. As a result of visualization, the road boundary line Pedestrians trespassing were detected according to the location of the pedestrian. However, the road boundary line detection method applied in this paper also has a problem in that it cannot create areas such as intersections or alleys except for straight lanes.

We are through this study, it is expected that it will be possible to prevent many fatal accidents by effectively detecting jaywalking in the intelligent traffic system, detecting risk factors, and using it in the ITS, an intelligent traffic system.

REFERENCES

[1] *http://taas.koroad.or.kr/*

[2] M. -K. Ba and S. -H. Park, “A Study on the Street Environmental Characteristics Influencing Jaywalking Traffic Accidents,” *Journal of The Korean Regional Development Association*, Vol. 30, No. 2, pp. 177-191, June 2018.

[3] J. -H. Kim and Y. -H. Shin, “A Study on Deep Learning-based Pedestrian Detection and Alarm System,” *The Journal of The Korea Institute of Intelligent Transport Systems*, Vol. 18, No. 4, pp. 58-70, Aug 2019. DOI: https://doi.org/10.12815/kits.2019.18.4.58

[4] Joseph Redmon and Ali Farhadi, "YOLO9000: better, faster, stronger," in *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 7263-7271, July, 2017. DOI: https://doi.org/10.48550/arXiv.1612.08242

[5] Redmon, Joseph and Ali Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, Apr 2018. DOI: https://doi.org/10.48550/arXiv.1804.02767

[6] Alexey Bochkovskiy, Chien-Yao Wang and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, Apr 23, 2020. DOI: https://doi.org/10.48550/arXiv.2004.10934

[7] H. T. Kim and S. H. Lee, “A study on object distance measurement using OpenCV-based YOLOv5,” *International Journal of Advanced Culture Technology*, Vol. 9, No. 3, pp.298-304, 2021. DOI : https://doi.org/10.17703/IJACT.2021.9.3.298

[8] *https://en.wikipedia.org/wiki/Canny_edge_detector*

[9] *https://AIHub.or.kr/aidata/34120*

[10] J. -H. Yu, Y. -J. Han and H. -S. Hahn, “On-road Vehicle and Area Detection Using Edge Connectivity and Corner Clustering,” *The Institute of Electronics and Information Engineers*, pp. 1035-1036, June 2008.

[11] J. -D. Oh, *A study on the lane detection techniques for intelligent vehicle*, MS Thesis. Sangji University, Gangwon-do, South Korea, 2016.