J Inf Process Syst, Vol.20, No.6, pp.827~840, December 2024
https://doi.org/10.3745/JIPS.04.0329
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Development of an Electric Scooter Photo Recognition System Using YOLO

Chaehyeon Kim, Sara Yu, and Ki Yong Lee*

Abstract

The use of electric scooter (e-scooter) sharing services has increased significantly in recent years due to their convenience and economy. In order to rent an e-scooter, a user first finds nearby e-scooters using a smartphone application, which shows the global positioning system (GPS) locations of e-scooters around the user. However, since the error of GPS can be more than 10 m, the user may have difficulty finding the exact location of the e-scooter the user wants to use. To alleviate this problem, an e-scooter sharing service "Kickgoing," operated by Olulo in South Korea, provides users with e-scooter photos taken by users upon return, along with their GPS locations, on its smartphone application. Those photos help subsequent users to find e-scooters more accurately. However, since some users upload photos that do not include e-scooters or are unrecognizable, it is essential to provide users with only those photos that clearly include an e-scooter. Therefore, in this paper, we develop an e-scooter photo recognition system that can accurately recognize only those photos that include e-scooters. The developed system, which is based on YOLO, uses three techniques: if a whole e-scooter is not recognized, it recognizes an e-scooter by recognizing its parts individually; it recognizes e-scooters with significantly different photography angles as different classes; and it provides users with only those photos in which the proportion of the e-scooter is within a certain range. Experimental results on a real dataset show that the developed system recognizes e-scooter photos more accurately compared to a system that uses the YOLO model as is.

Keywords

Electric Scooter Sharing Service, Object Detection, Photo Recognition, YOLO

1. Introduction

In recent years, the use of electric scooter (e-scooter) sharing services has increased significantly because they can contribute to improving urban problems such as traffic congestion, air pollution, and lack of parking space, as well as providing convenience and economy [1]. In order to rent an e-scooter, a user first finds the locations of nearby e-scooters using a smartphone application, which shows the GPS locations of e-scooters around them, and then move to the location of the e-scooter the user wants to use. However, since global positioning system (GPS) locations contain some errors, especially in places with many tall buildings, the actual location of an e-scooter may differ by more than 10 m. In this case, the user may have difficulty finding the exact location of the e-scooter the user wants to use.

Manuscript received February 23, 2023; first revision May 30, 2023; accepted June 25, 2023.
***Corresponding Author:** Ki Yong Lee (kiyonglee@sookmyung.ac.kr)
Dept. of Computer Science, Sookmyung Women's University, Seoul, Korea ({7chaeny25, rrrr4ra, kiyonglee}@sookmyung.ac.kr)
Current affiliation for author, Chaehyeon Kim, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, PA, USA.

To address the difficulty of finding an e-scooter only with a GPS location, an e-scooter sharing service "Kickgoing" [2], the South Korea's first e-scooter sharing service operated by Olulo Co. Ltd., asks users to take and upload a photo of the e-scooter they used upon return. Using these photos, Kickgoing's smartphone application provides users with not only the GPS locations of e-scooters but also their photos taken by previous users. Consequently, subsequent users can find an e-scooter more accurately and easily using these photos. However, some users upload photos that do not include e-scooters or are unrecognizable. For example, although most users upload photos that clearly include an e-scooter as in Fig. 1(a), some photos, like Fig. 1(b), do not include an e-scooter or are difficult to recognize an e-scooter. Therefore, in order to provide only useful information to users, it is very important to provide users with only those photos where an e-scooter is clearly recognizable [3]. However, since the shape of an e-scooter varies greatly depending on the angle from which its photo was taken, as shown in Fig. 1(a) and 1(c), recognizing an e-scooter is very challenging.

(a) (b) (c)

Fig. 1. Examples of e-scooter photos taken and uploaded by actual users upon return: (a) a photo that clearly includes an e-scooter, (b) a photo that is difficult to recognize an e-scooter, and (c) a photo that includes an e-scooter of a very different shape.

Therefore, in this paper, we develop an e-scooter photo recognition system that can accurately recognize only those photos that include an e-scooter. The developed system is based on YOLO, a state-of-the-art object detection model [4]. We use the following three techniques to develop the system:

- We first try to recognize an e-scooter as a whole. If a whole e-scooter is not recognized, then we divide an e-scooter into three parts (i.e., a handle, a bar, and a bottom) and try to recognize each part individually. If all the three parts are recognized and their bounding boxes overlap, we consider an e-scooter recognized. In this way, we can recognize more e-scooter photos effectively.
- We treat e-scooters with significantly different photography angles as different classes and recognize them as different classes. More specifically, we label e-scooters in photos as different classes according to the angle between their bar and bottom parts (e.g., $0° \le \theta < 60°$, $60° \le \theta < 120°$, and $120° \le \theta < 180°$). Then, if a photo is classified into any of these classes, we consider an e-scooter recognized. In this way, we can more accurately recognize e-scooters that have very various shapes.
- Finally, we consider an e-scooter recognized in a photo only when the proportion of the e-scooter in the photo is within a certain range (e.g., 10% to 70%). If an e-scooter in a photo is too small or too large, a user cannot get enough information about where the e-scooter is currently located. Therefore, we provide users with only those photos in which the proportion of the e-scooter is within a pre-specified range. In this way, we can provide only useful photos to users.

Through experiments using photos taken and uploaded by actual users, we show that the developed system can recognize e-scooter photos more accurately and effectively compared to a system that uses the YOLO model as is.

The rest of the paper is organized as follows: Section 2 reviews representative studies on deep learning-based object detection algorithms, and Section 3 describes our e-scooter photo recognition system in detail. Section 4 presents experimental results on a real dataset to show the effectiveness of our developed system. Finally, Section 5 concludes the paper.

2. Related Work

Object detection is a computer vision technology that identifies objects on an image or video that distinguishes objects of interest from the background [5]. Object detection is used in various fields, such as autonomous driving (to recognize stop signals or distinguish pedestrians from streetlights), bio-imaging (to identify diseases from an image), and people counting (to track the number of persons in a place). The problem of this paper can be expressed as an e-scooter detection problem. In general, given an image, an object detection algorithm outputs the classes of the detected objects in the image and the bounding boxes surrounding them. Currently, deep learning models are used as the state-of-the-art algorithms for object detection. Representative deep learning-based object detection algorithms are divided into two-step algorithms and one-step algorithms.

2.1 Deep Learning-based Two-Step Object Detection Algorithms

Most deep learning-based two-step object detection algorithms consist of a region proposal step and an object detection step. In the region proposal step, the algorithm generates candidate regions in the image to detect objects. Next, in the object detection step, the algorithm performs classification on each candidate region to detect objects in that region. Currently, faster region-based convolutional neural network (Faster R-CNN) [6] is a state-of-the-art two-step object detection algorithm, whose architecture is shown in Fig. 2. As the name of the algorithm implies, Faster R-CNN is faster than the previous versions of the R-CNN algorithm [7,8]. In the region proposal step, Faster R-CNN uses a region proposal network to generate candidate regions that are likely to contain objects, from the feature map produced by the convolutional layers. Because the region proposal network is simply performed on the feature map produced by the convolutional layers, the region proposal step is nearly cost-free. In the object detection step, Faster R-CNN extracts the feature map corresponding to each candidate region from the feature map produced by the convolutional layers and performs classification on the extracted feature map to predict the objects included in that candidate region. Compared to the previous versions, Faster R-CNN is very fast because it does not need to scan the entire image.

2.2 Deep Learning-based Single-Step Object Detection Algorithms

You Only Look Once (YOLO) [9] is a representative deep learning-based single-step object detection algorithm. YOLO divides the original image into grids of equal size. Given an input image, YOLO predicts objects, their bounding boxes, and their confidences for all grids at once, which is why YOLO

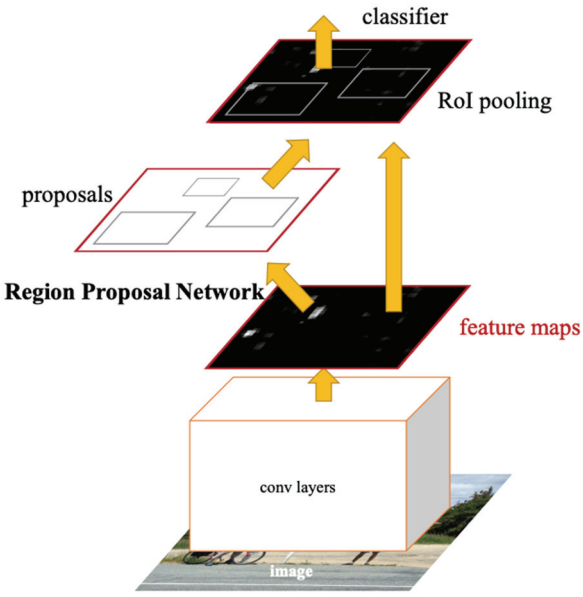


Fig. 2. The architecture of Faster R-CNN [6].

is called a single-step algorithm. If the same object is detected on multiple grids, the grid with the highest confidence is selected for that object. Also, for each grid, YOLO can detect multiple objects, each of which has a bound box with a different predefined shape. Each bounding box with a different predefined shape is called an "anchor box." Typically, the shapes of anchor boxes are obtained by grouping the shapes of objects using a clustering algorithm such as *k*-means. Each anchor box is intended to detect objects of a different shape. If there are multiple objects with different shapes within a grid, YOLO detects each object using a different anchor box. Fig. 3 shows the architecture of the YOLO model. The CNN layers extract features from an image and the detection layer outputs the predicted objects, their bounding boxes, and their confidences for each grid. There are several versions of YOLO [10], each of which comes in four models with different sizes: small (s), medium (m), large (l), and extra large (x). The larger the model size, the higher the accuracy, but the more parameters and the longer the training time. YOLO is known to be faster than Faster R-CNN in general. This is because Faster R-CNN needs to perform classification on each candidate region, whereas YOLO needs to perform only one classification to make predictions for all grids.

Although various deep learning-based object detection algorithms have been proposed, applying them directly to our e-scooter photo recognition problem does not give the best performance. This is because, especially in the case of an e-scooter, its shape varies greatly depending on the angle from which its photo was taken, as shown in Fig. 1(a) and 1(c). As more examples, if a person takes a photo of an e-scooter from the front, the e-scooter would look like an "I," and if the person takes a photo of the e-scooter from the left side, the e-scooter would look like an "L." Therefore, we need a system that can recognize e-scooter photos more accurately and effectively. As far as we know, our earlier work [11] is the only work on improving the accuracy of e-scooter recognition by dividing e-scooters into parts and angles. However, [11] has a limitation in that it uses only a simple method to recognize an e-scooter by its parts. This is the motivation of our work.

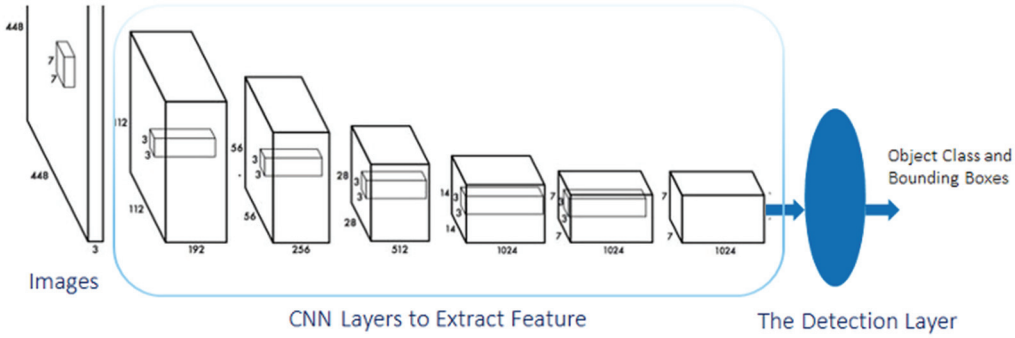


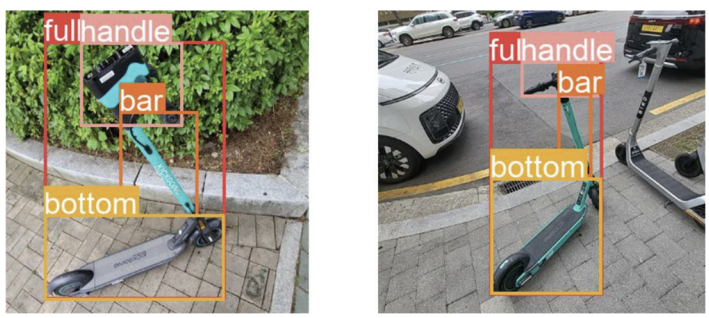
Fig. 3. The architecture of YOLO [9].

3. Our E-Scooter Photo Recognition System

In this paper, we develop a YOLO-based e-scooter photo recognition system that can recognize e-scooter photos more accurately and effectively. We chose to use YOLO because it is fast, open source, and provides adequate performance [12]. In order to accurately and effectively recognize e-scooters, whose shape varies greatly depending on the angle from which their photos were taken, we use three techniques: (1) if we fail to recognize an e-scooter as a whole, then we try to recognize an e-scooter by recognizing its parts individually, (2) we label e-scooters as different classes according to the angle between their bar and bottom parts, and then consider an e-scooter recognized if the photo is classified into any one of these classes, and (3) we consider an e-scooter recognized only when the proportion of the e-scooter in the photo is within a certain range. In the next subsections, we describe each technique in detail.

3.1 Part-based E-Scooter Recognition

An e-scooter can be roughly divided into three parts: a handle, a bar, and a bottom, as shown in Fig. 4. Although the overall shape of an e-scooter varies greatly depending on the angle from which its photo was taken, each of the three part maintains a relatively consistent shape. Therefore, we label each of the three parts individually as shown in Fig. 4 and use them for e-scooter photo recognition.



(a) (b)

Fig. 4. (a, b) Examples of part labeling.

Given an input photo, we first try to recognize an e-scooter as a whole. If we fail to recognize an e-scooter as a whole, we try to recognize all the three parts of an e-scooter (i.e., a handle, a bar, and a bottom) individually. Fig. 5 shows this process. If all the three parts are recognized successfully and their bounding boxes overlap, then we consider an e-scooter recognized and output the minimum bounding box that encloses the bounding boxes of the three parts.

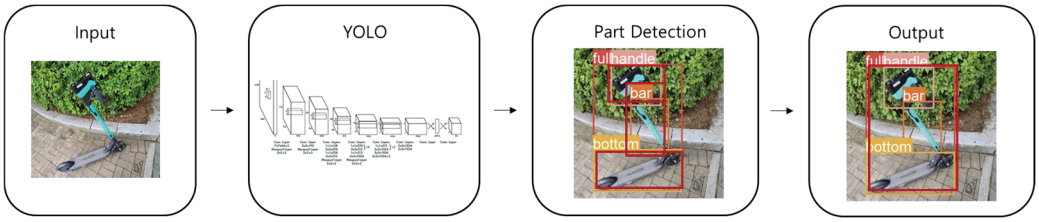


Fig. 5. The overall process of part-based e-scooter recognition.

Let c_{handle}, c_{bar}, and c_{bottom} be the confidence of the handle, bar, and bottom of an e-scooter predicted by YOLO, respectively. Note that the values of c_{handle}, c_{bar}, and c_{bottom} can be different significantly. Based on the values of c_{handle}, c_{bar}, and c_{bottom}, we need to determine whether an e-scooter is recognized or not. We use the following three strategies to determine whether an e-scooter is recognized or not:

Strategy 1: We consider an e-scooter recognized if $c_{handle} \geq M$, $c_{bar} \geq M$, and $c_{bottom} \geq M$, where M is the predetermined minimum confidence. In other words, we consider an e-scooter recognized only when all of the three parts are recognized with the same level of confidence or higher. Note that if any of the three parts are not recognized with sufficient confidence, we consider an e-scooter not recognized. M is a hyperparameter, which can be determined by a validation set.

Strategy 2: We consider an e-scooter recognized if $c_{handle} \geq M_{handle}$, $c_{bar} \geq M_{bar}$, and $c_{bottom} \geq M_{bottom}$, where M_{handle}, M_{bar}, and M_{bottom} are the minimum confidence for the handle, bar, and bottom, respectively. Different from Strategy 1, this strategy uses different minimum confidences for different parts. This strategy can be more effective than Strategy 1 if the handle, bar, and bottom have different recognition difficulties. For example, in the experiments on a real dataset, we observed that bars are more difficult to recognize than handles and bottoms. In this strategy, M_{handle}, M_{bar}, and M_{bottom} are hyperparameters to be determined by a validation set.

Strategy 3: We consider an e-scooter recognized if $w_{handle} \cdot c_{handle} + w_{bar} \cdot c_{bar} + w_{bottom} \cdot c_{bottom} \geq M$, where w_{handle}, w_{bar}, and w_{bottom} are the weights of c_{handle}, c_{bar}, and c_{bottom}, respectively such that $w_{handle} + w_{bar} + w_{bottom} = 1$, and M is the minimum confidence. This strategy uses the weighted average of c_{handle}, c_{bar}, and c_{bottom} as the criterion instead of using them individually. In this strategy, w_{handle}, w_{bar}, w_{bottom}, and M are hyperparameters to be determined by a validation set.

These strategies use different criteria and hyperparameters for recognizing an e-scooter, and thus have different recognition performance. In Section 4, we compare the performance of these strategies for various hyperparameter values.

3.2 Angle-based E-Scooter Recognition

As mentioned previously, the shape of an e-scooter can vary significantly depending on the angle from

which its photo was taken. For example, as shown in Fig. 6, if a photo is taken from the side of an e-scooter, the e-scooter would look like an “L” shape, and if a photo is taken from the front or back of an e-scooter, the e-scooter would look like an “I” shape. Since the shapes of e-scooters vary greatly from photo to photo, it is very difficult for YOLO to learn that they all belong to the same class.

As a result, the recognition performance may suffer if we train the YOLO model by labeling all e-scooters with various shapes as the same class. To address this problem, we label e-scooters as different classes according to the angle between their bar and bottom. Let θ be the angle between the bar and bottom of a given e-scooter in a photo. According to θ, we label the e-scooter as follows:

If $0° \leq \theta < 60°$, then we label the e-scooter as the class, Angle1 (e.g., Fig. 6(a)).
If $60° \leq \theta < 120°$, then we label the e-scooter as the class, Angle2 (e.g., Fig. 6(b)).
If $120° \leq \theta < 180°$, then we label the e-scooter as the class, Angle3 (e.g., Fig. 6(c)).

(a) (b) (c)

Fig. 6. Examples of e-scooter photos taken from various angles: (a) a photo where $0° \leq \theta < 60°$, (b) a photo where $60° \leq \theta < 120°$, and (c) a photo where $120° \leq \theta < 180°$.

Then we train YOLO on a dataset labeled in this way. This enables YOLO to learn the shapes of e-scooters more accurately for each range of θ. After training, if an object is classified into any of the classes, Angle1, Angle2, and Angle3, we consider an e-scooter recognized. Fig. 7 shows this process. We will present experimental results on the effectiveness of this technique in Section 4.2.2.

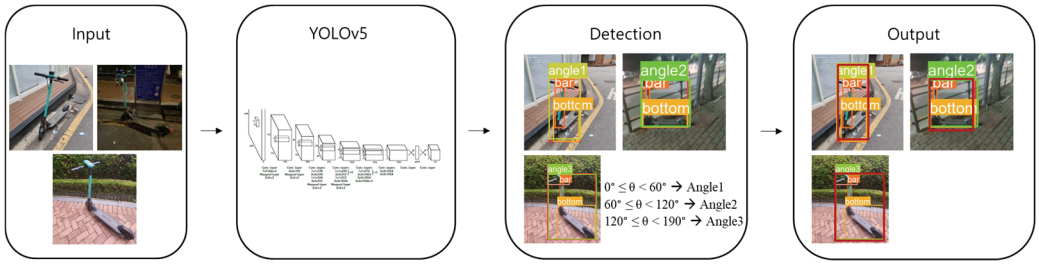


Fig. 7. The overall process of angle-based e-scooter recognition.

3.3 Proportion-based E-Scooter Recognition

As shown in Fig. 8, users can upload e-scooter photos taken at various distances. However, if an e-scooter in a photo is too small or too large, the user cannot get enough information about where the e-scooter is currently located. For example, if an e-scooter in a photo is too large, like Fig. 8(c), a user cannot easily identify the surrounding environment where the e-scooter is currently located. Therefore,

we consider an e-scooter recognized only when the proportion of the e-scooter in the photo is within a certain range (e.g., 10% to 90%). That is, among the photos where an e-scooter is recognized, we provide users with only those photos in which the proportion of the e-scooter in the photo is within a pre-specified range. In this way, we can provide only useful photos to users.

(a) (b) (c)

Fig. 8. Examples of e-scooter photos taken at various distances: (a) a photo in which the proportion of an e-scooter is less than 10%, (b) a photo in which the proportion of an e-scooter is about 60%, and a photo in which the proportion of an e-scooter is about 90%.

Fig. 8 shows examples of e-scooter photos taken at various distances. Fig. 8(a) shows a photo in which the proportion of an e-scooter is less than 10%. On the other hand, Fig. 8(b) and 8(c) show a photo in which the proportion of an e-scooter is about 60% and 90%, respectively. As can be inferred from these photos, if the proportion of an e-scooter in a photo is less than 10% or greater than 90%, the photo is unlikely to provide useful information for users to find the exact location of the e-scooter. Accordingly, we only provide users with e-scooter photos in which the proportion of an e-scooter is between 10% and 90%.

4. Experiments

In this section, we present experimental results on how accurately and effectively our developed system can recognize e-scooter photos.

4.1 Experimental Setting

We implemented our e-scooter photo recognition system in PyTorch using YOLOv5 [13] as the base model. We conducted experiments on a PC running Windows 10 equipped with an Intel Core i7-9700 3.3 GHz GPU and 16 GB RAM. In order to train and test our system, we used a real dataset provided by Olulo Co. Ltd., which consists of 3,102 photos taken and uploaded by real Kickgoing users. The real dataset contains a mix of photos with and without e-scooters. This dataset has only one-class, which indicates whether each photo includes an e-scooter or not. For the experiments, we split the dataset into a training set, a validation set, and a test set, which contain 70%, 20%, and 10% of the dataset, respectively. In order to evaluate the performance of our e-scooter photo recognition system, we compared the performance of the following systems.

- BASELINE: A system that uses the YOLOv5 model as is without any additional techniques

- PART_BASED: A system that uses the part-based e-scooter recognition technique described in Section 3.1
- ANGLE_BASED: A system that uses the angle-based e-scooter recognition technique described in Section 3.2
- PROPOSED: A system that uses both the part-based and angle-based e-scooter recognition techniques. More specifically, this system first applies the angle-based e-scooter recognition technique to recognize an e-scooter in a photo. If the recognition fails, then the system applies the part-based e-scooter recognition technique to additionally recognize an e-scooter.

Note that we did not evaluate the performance of the proportion-based e-scooter recognition technique described in Section 3.3, because it is not a technique to increase the accuracy but to provide more informative photos to users. To compare the performance of the above systems, we measured the precision, recall, and F1 score of the systems. In the next subsection, we present experimental results on the performance of the above systems.

4.2 Experimental Results

We first present the performance evaluation results of the part-based e-scooter recognition technique described in Section 3.1. Table 1 compares the performance of BASELINE and PART_BASED. For PART_BASED, as described in Section 3.1, we can use three strategies to determine whether an e-scooter is recognized or not from the values of c_{handle}, c_{bar}, and c_{bottom}. Table 1 shows the performance of PART_BASED when each strategy is use.

Table 1. Performance evaluation results of the part-based e-scooter recognition technique

System	Performance		
	Precision	**Recall**	**F1-score**
BASELINE	0.973	0.831	0.896
PART_BASED			
Strategy 1	0.976	0.963	0.969
Strategy 2	0.976	0.978	0.977
Strategy 3	0.977	0.993	0.985

As shown in Table 1, PART_BASE shows superior performance compared to BASELINE regardless of its strategy. This means that our part-based e-scooter recognition technique can improve the performance of BASELINE by additionally recognizing e-scooters that are difficult to recognize as a whole at once. Note also that the performance of PART_BASED with Strategy 2 and 3 are better than that with Strategy 1. As described in Section 3.1, Strategy 1 uses the same minimum confience M for all of c_{handle}, c_{bar}, and c_{bottom}. In comparison, Strategy 2 and 3 use a more flexible criterion that allows a different minimum confidence for each of c_{handle}, c_{bar}, and c_{bottom}. As a result, PART_BASED with Strategy 2 and 3 show better performance than PART_BASED with Strategy 1. However, note that Strategy 2 and 3 require more hyperpameters than Strategy 1.

In Table 1, the performance of PART_BASED with each strategy is obtained by searching for the best hyperparameters by grid search. Tables 2–4 show the top 5 performance of PART_BASED with Strategy 1, 2, and 3, respectively, for various values of the hyperparameters.

Table 2. Performance of PART_BASED with Strategy 1 for various hyperparameters (top 5)

Hyperparameter	Performance		
	Precision	Recall	F1 Score
$M = 0.5$	**0.976**	**0.963**	**0.969**
$M = 0.6$	0.976	0.951	0.963
$M = 0.7$	0.975	0.935	0.955
$M = 0.8$	0.975	0.912	0.942
$M = 0.9$	0.973	0.862	0.914

The bold font indicates the best performance.

Table 3. Performance of PART_BASED with Strategy 2 for various hyperparameters (top 5)

Hyperparameter			Performance		
M_{handle}	M_{bar}	M_{bottom}	Precision	Recall	F1-score
0.4	**0.4**	**0.45**	**0.976**	**0.978**	**0.977**
0.45	0.4	0.45	0.976	0.976	0.976
0.4	0.4	0.5	0.976	0.975	0.975
0.5	0.4	0.45	0.976	0.968	0.972
0.5	0.45	0.45	0.976	0.968	0.972

The bold font indicates the best performance.

Table 4. Performance of PART_BASED with Strategy 3 for various hyperparameters (top 5)

Hyperparameters				Performance		
w_{handle}	w_{bar}	w_{bottom}	M	Precision	Recall	F1-score
0.3	**0.25**	**0.45**	**0.4**	**0.977**	**0.993**	**0.985**
0.35	0.3	0.35	0.4	0.977	0.991	0.983
0.33	0.33	0.33	0.4	0.977	0.986	0.981
0.3	0.25	0.45	0.5	0.977	0.985	0.981
0.33	0.33	0.33	0.5	0.976	0.967	0.971

The bold font indicates the best performance.

Next, we present the performance evaluation results of the angle-based e-scooter recognition technique described in Section 3.2. Table 5 compares the performance of BASELINE and ANGLE_BASED. In this experiment, we split the dataset into a training set, a validation set, and a test set, which contain 50%, 30%, and 20% of the dataset, respectively, in order to use more data as a test set.

Table 5. Performance evaluation results of the angle-based e-scooter recognition technique

System	Performance		
	Precision	Recall	F1-score
BASELINE	0.826	0.888	0.856
ANGLE_BASED	0.895	0.905	0.9

As shown in Table 5, ANGLE_BASED provides better performance than BASELINE. Recall that the shapes of e-scooters vary greatly depending on the angle from which their photo is taken. By dividing e-

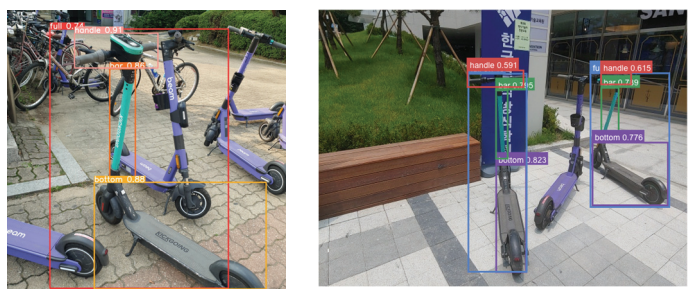
scooter photos into three groups according to the angle between the bar and bottom of e-scooters ($0° \leq \theta < 60°$, $60° \leq \theta < 120°$, and $120° \leq \theta < 180°$) and learning each group as a different class, ANGLE_BASED can recognize e-scooters of various shapes more accurately. From the experimental results in Table 5, we can confirm that the angle-based e-scooter photo recognition is more effective than learning e-scooters of various shapes as a single class.

Finally, Table 6 compares the performance of BASELINE, PART_BASED, ANGLE_BASED, and PROPOSED. Note that PROPOSED represents the final system we developed, which uses both of the two techniques, i.e., the part-based and angle-based e-scooter photo recognition. In this experiment, we split the dataset into a training set, a validation set, and a test set, which contain 70%, 20%, and 10% of the dataset, respectively. From Table 6, we can see that PROPOSED provides the best performance among the four systems. Therefore, we can confirm that the proposed e-scooter photo recognition techniques can be used effectively to improve the performance of recognizing e-scooter photos.

Table 6. Performance comparison of the four systems

System	Performance		
	Precision	**Recall**	**F1-score**
BASELINE	0.973	0.831	0.896
PART_BASED	0.977	0.993	0.985
ANGLE_BASED	1	0.847	0.917
PROPOSED	**1**	**0.998**	**0.999**

The bold font indicates the best performance.



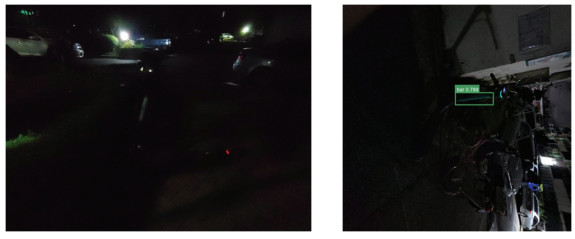
(a) (b)

Fig. 9. Examples of photos with e-scooters from various companies: (a) a photo where there are e-scooters from other companies and (b) another photo where there are e-scooters from other companies.

Note that there can be e-scooters from other companies that look similar in the same photo. However, they usually have different characteristics (e.g., color and written letters) than Olulo's e-scooter. Because the proposed system only learns the characteristics of Olulo's e-scooters, the proposed system can accurately recognize only Olulo's e-scooters even when there are e-scooters from other companies in the same photo. Fig. 9 shows examples of photos where there are e-scooters from another company. However, even in these cases, the proposed system can accurately recognize only Olulo's e-scooters by detecting their mint-colored bars.

However, the proposed system still has a limitation that its recognition performance is degraded for photos taken in low-light conditions. As shown in Fig. 10(a), if a photo is taken in a very dark

environment, the proposed system may not properly recognize an e-scooter in the photo. However, in the case of Fig. 10(b), although the photo is dark, the propose system can recognize the e-scooter by detecting its bar. In contrast, BASELINE cannot recognize the e-scooter in Fig. 10(b) because it recognizes an e-scooter as a whole. Thus, even for photos taken in low-light conditions, the proposed system can improve the recognition performance by detecting the parts of e-scooters individually, although the overall recognition performance is degraded.



(a) (b)

Fig. 10. Examples of photos taken in low-light conditions: (a) a photo taken in a very dark environment and (b) a photo where the bar of an e-scooter is detected.

5. Conclusion

In this paper, we developed a YOLO-based e-scooter photo recognition system. Compared to a system that simply uses the YOLO model as is without any additional techniques, our e-scooter photo recognition system can recognize e-scooter photos more accurately and effectively. Because the shapes of e-scooters can vary considerably depending on the angle and distance from which their photo was taken, our system uses three techniques: (1) if an e-scooter is not recognized as a whole, we try to recognize the parts of an e-scooter individually and combine these recognition results to additionally recognize an e-scooter, (2) we divide e-scooter photos into several groups according to the angle between the bar and bottom of the e-scooter, and then treat each group as a different class when training and testing the YOLO model, and (3) we consider an e-scooter photo recognized only when the proportion of the e-scooter in the photo is not too small and not too large in order to provide users with only photos that are useful for locating an e-scooter. The experimental results on the real dataset show that our developed system improves the performance of recognizing e-scooter photos significantly.

Conflict of Interest

The authors declare that they have no competing interests.

Funding

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1A2C1012543).

Acknowledgments

This paper is the extended version of "Development of a YOLO-Based Electric Kick Scooter Photo Recognition System," in the Annual Conference of KIPS (ACK 2022) held in Seoul, Republic of Korea dated November 3–5, 2022.

References

[1] S. J. Kim, G. J. Lee, S. Choo, and S. H. Kim, "Study on shared e-scooter usage characteristics and influencing factors," *The Journal of The Korea Institute of Intelligent Transport Systems*, vol. 20, no. 1, pp. 40-53, 2021. https://doi.org/10.12815/kits.2021.20.1.40

[2] Kickgoing [Online]. Available: https://kickgoing.io/.

[3] S. Gilroy, D. Mullins, E. Jones, A. Parsi, and M. Glavin, "E-scooter rider detection and classification in dense urban environments," 2022 [Online]. Available: https://arxiv.org/abs/2205.10184.

[4] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A review of Yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066-1073, 2022. https://doi.org/10.1016/j.procs.2022.01.135

[5] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object detection with deep learning: a review," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, 2019. https://doi.org/10.1109/TNNLS.2018.2876865

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," 2015 [Online]. Available: https://arxiv.org/abs/1506.01497v1.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, 2014, pp. 580-587. https://doi.org/10.1109/CVPR.2014.81

[8] R. Girshick, "Fast R-CNN," in *Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015, pp. 1440-1448. https://doi.org/10.1109/ICCV.2015.169

[9] A. Aralikatti, J. Appalla, S. Kushal, G. S. Naveen, S. Lokesh, and B. S. Jayasri, "Real-time object detection and face recognition system to assist the visually impaired," *Journal of Physics: Conference Series*, vol. 1706, no. 1, article no. 012149, 2020. https://doi.org/10.1088/1742-6596/1706/1/012149

[10] T. Diwan, G. Anirudh, and J. V. Tembhurne, "Object detection using YOLO: challenges, architectural successors, datasets and applications," *Multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243-9275, 2023. https://doi.org/10.1007/s11042-022-13644-y

[11] C. Kim, S. Yu, S. Yoon, G. Kim, H. Kong, J. Lee, S. Song, and K. Y. Lee, "Development of a YOLO-based electric kick scooter photo recognition system," in *Proceedings of Annual Conference of the Korea Information Processing Society (ACK)*, 2022, pp. 622-624. https://doi.org/10.3745/PKIPS.y2022m11a.622

[12] H. Kim, M. K. Sohn, and S. H. Lee, "Development of a real-time automatic passenger counting system using head detection based on deep learning," *Journal of Information Processing Systems*, vol. 18, no. 3, pp. 428-442, 2022. https://doi.org/10.3745/JIPS.04.0246

[13] G. Jocher, "YOLO5," 2020 [Online]. Available: https://github.com/ultralytics/yolov5.

Chaehyeon Kim **https://orcid.org/0000-0002-6443-1286**

She received her B.S. in both Statistics and Computer Science from Sookmyung Women's University, Seoul, Republic of Korea, in 2021. She is now an M.S. student in the Department of Computer Science degrees at Sookmyung Women's University, Seoul, Korea. Her current research interests include databases, data mining, and deep learning.

Sara Yu **https://orcid.org/0000-0002-1294-8320**

She received her B.S. in Computer Science from Sookmyung Women’s University, Seoul, Republic of Korea, in 2022. She is now an M.S. student in the Department of Computer Science of degrees at Sookmyung Women's University, Seoul, Korea. Her current research interests include databases, data mining, big data processing and deep learning.

Ki Yong Lee **https://orcid.org/0000-0003-2318-671X**

He received his B.S., M.S., and Ph.D. degrees in Computer Science from KAIST, Daejeon, Republic of Korea, in 1998, 2000, and 2006, respectively. From 2006 to 2008, he worked for Samsung Electronics, Suwon, Korea as a senior engineer. From 2008 to 2010, he was a research assistant professor of the Department of Computer Science at KAIST, Daejeon, Korea. He joined the faculty of the Division of Computer Science at Sookmyung Women’s University, Seoul, in 2010, where currently he is a professor. His research interests include database systems, data mining, and big data.