ESKİŞEHİR TECHNICAL UNIVERSITY JOURNAL OF SCIENCE AND TECHNOLOGY
RESEARCH ARTICLE – COMPUTER ENGINEERING AND COMPUTER SCIENCE

2022 December

CROWDED BUS STOP SOLUTION

YUNUS EMRE KORKMAZ[1*], UMUTCAN YAVUZ[2*]

[1] Computer Engineering, Engineering Faculty, Eskisehir Technical University, Eskisehir, Turkey.
[2] Computer Engineering, Engineering Faculty, Eskisehir Technical University, Eskisehir, Turkey.

ABSTRACT

In the globalizing world, the population of people is increasing day by day. These increases also cause some transportation problems. The first of these is buses used by almost everyone. Every day, millions of people wasting their time waiting at bus stops. In this scenario, we think a solution to decrease the waiting time of people with using object detection algorithms. In this article, we will show how to obtain the data and what can we do with those data to solve this problem. We trained the yolov5 with coco [2] data and process the image inputs with yolov5[1] object detection library and get the total count of waiting people on a bus stop. This is showing us this solution is applicable and feasible with help of cameras that placed in bus stops. In the future, this idea can be using for decreasing the waiting time of people on bus stops.

Keywords: Object detection, Video Recording, Public bus stops, Waiting time decrease

1. INTRODUCTION

Every day we waste our time waiting at bus stops. With our solution we can decrease the waiting time with object detection algorithms.

The solution that we used for this problem is simply object detection and counting the people from videos or images that we will obtain from cameras that placed top of bus stops. The images will be bird's eye view. When we get the inputs, we will process this data with yolov5[1] object detection libraries trained with --- data. At the end of this part, we will obtain the data of waiting people count on a certain time. The data will be represented with a time stamp. With this information we can predict the future time's waiting people count on a certain time. We can inform bus administrators with that data so they can send the required buses to that bus stop. Thus, we can decrease the waiting time of people waiting on bus stops.

If we placed the cameras and take video inputs on 7/24 from a bus stop with bird's eye view like on the below figure (Figure 1.). We can split this to parts according to different busses (Bus1, Bus2, Bus3).

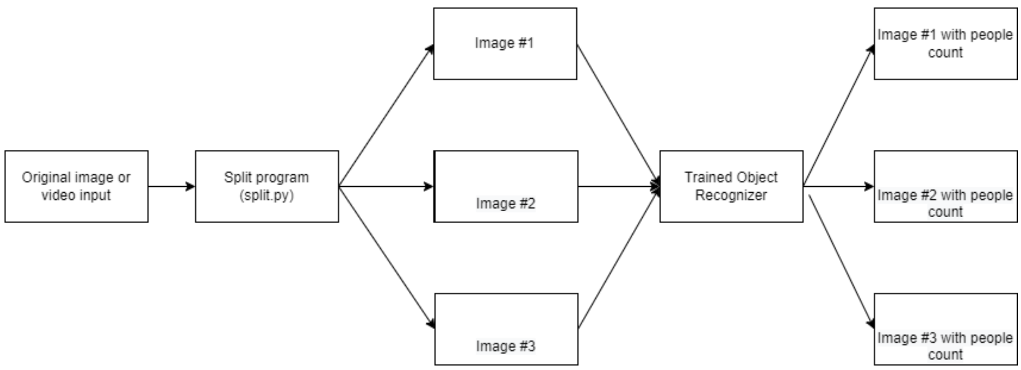

Figure 1. The above figure represents the main image's split phase and people count recognition phase

We trained our program with coco dataset [2]. After that train part, we process the images and videos and get the results. We concluded that; we can use this data to generate tables of outputs as you can see on following tables on every bus stop:

Table 1. Waiting people count for a bus "B" from the camera on a bus stop with id 1.

time/buses	B_1	B_2	B_3	STOP ID
T_1	3	2	5	1
T_2	4	4	5	1
T_3	7	4	9	1
.	.	.	.	1
.	.	.	.	1
.	.	.	.	1
T_n	.	.	.	1

Table 2. Waiting people count for a bus "B" from the camera on a bus stop with id 2.

time/buses	B_1	B_2	B_3	STOP ID
T_1	3	2	5	2
T_2	4	4	5	2
T_3	7	4	9	2
.	.	.	.	2
.	.	.	.	2
.	.	.	.	2
T_n	.	.	.	2

Table 3. Total count of waiting people for a bus B on a time T.

time/stops	B_1	B_2	B_3
T_1	.	.	.
T_2	.	.	.
T_n			

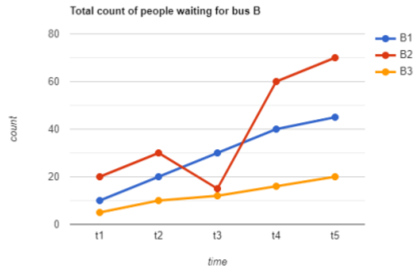

Figure 2. The above figure represents the graph of total count of people waiting for bus 'B' in each time 'T'

As we can see from table 1, 2, 3 and figure 2. it is obvious that we can achieve to get the total waiting people count for a bus on a given time.

2. MATERIALS

We use the following dataset to train our model with yolov5[1]:

1. Coco Dataset [1]

Objects are labeled using per-instance segmentations to aid in precise object localization. The dataset contains photos of 91 object types that would be easily recognizable by a 4-year-old. Dataset contains a total of 2.5 million labeled instances in 328k images. [4]

3. METHODS

We use object detection algorithms from yolov5[1] with the datasets that we mention at materials part.

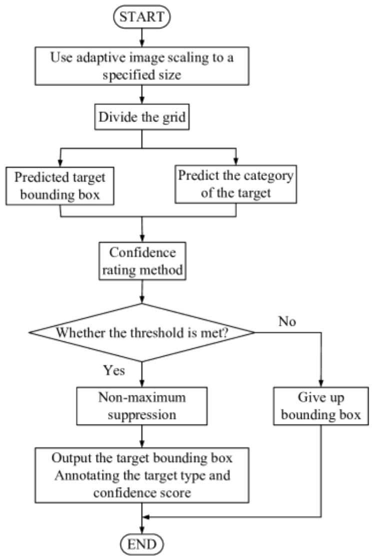

Figure 3[5]. The above figure represents the flowchart of yolov5[1]'s object detection.

3.1 FEATURE EXTRACTION

We use photos from different bus stops on different angles.

3.2 TRAINING

We trained the program with coco [2] dataset.

4. RESULTS

With object detection algorithm yolov5 [1], we got the count of waiting peoples for a bus. The examination that we have done clearly explains that this project is applicable and feasible for crowded cities along the world with proper infrastructure. Also, we can expand this project by using this data for calculating the future's people counts.

5. DISCUSSION

There are several problems that may occur during the object detection phase. For example, the images can contain the people that does not waiting for buses. Also, program does not work for %100 accuracy. There may be little bit errors on detection phase. Protection of personal data may also be a problem for this idea. Putting the cameras on every bus stop is increasing the cost of this project. Also, sending more bus-to-bus stops on every crowd may cause financial losses in terms of driver salaries and fuel.

CONTRIBUTIONS

In this project, Yunus Emre KORKMAZ, implemented the yolov5 [1] library on our Python [6] code and prepared the necessary environment for applying the required procedures. Also, he made a literature review from different articles.

Umutcan YAVUZ, gathered the appropriate inputs from bus stops on Google Maps [7] and different sources and tested it. He wrote the codes of image split script. Also, prepared the figures and tables.

CONFLICT OF INTEREST

The authors stated that there are no conflicts of interest regarding the publication of this article.

ETHICAL STATEMENT

While preparing the project, we did not take any action that could be considered as plagiarism.

REFERENCES

[1] Jocher, G. (2020). YOLOv5 by Ultralytics (Version 7.0). Zenodo. Retrieved from https://doi.org/10.5281/zenodo.3908559

[2] Tsung-Yi Lin, Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., … Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. CoRR, abs/1405.0312. Retrieved from http://arxiv.org/abs/1405.0312

[3] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248–255).

[4] Lin, TY. et al. (2014). Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48.

[5] Lu, Yao & Qiu, Zhibin & Liao, Caibo & Zhou, Zhibiao & Li, Tonghongfei & Wu, Zijian. (2022). A GIS Partial Discharge Defect Identification Method Based on YOLOv5. Applied Sciences. 12. 8360. 10.3390/app12168360.

[6] Van Rossum, G., & Drake Jr, F. L. (1995). Python reference manual. Centrum voor Wiskunde en Informatica Amsterdam.

[7] Google (2022) 15 Seylap Caddesi. Available at: https://www.google.com/maps/ (Accessed: 20 December 2022).