Multiple Objects Detection and Classification for Street Intersection Surveillance Video Based on Deep Learning

Ri-Chen Feng, Daw-Tung Lin and Yi-Yao Lin

February 22, 2019

MULTIPLE OBJECTS DETECTION AND CLASSIFICATION FOR STREET INTERSECTION SURVEILLANCE VIDEO BASED ON DEEP LEARNING

[1]*Ri-Chen Feng,* [2]*Daw-Tung Lin,* [3]*Yi-Yao Lin*

[1,2]Department of Computer Science and Information Engineering National Taipei University
New Taipei City, Taiwan
[3]Business Solutions Laboratory, Telecommunication Laboratories
Taoyuan, Taiwan
[1]s710583125@gm.ntpu.edu.tw, [2]dalton@mail.ntpu.edu.tw, [3]yylin@cht.com

ABSTRACT

This research uses deep learning techniques to classify image objects of intelligent image recognition from various public environments (e.g., intersections, campuses and community safe). After receiving an image, able to use quickly pre-trained several categories such as large cars, small cars, motorcycles, and bicycles to classification, using the “you only look once” deep learning architecture for training and detection. We filter and balance classes and quantities of input training data to ensure they can better model the images and improve detection stability. Therefore, the mAP of our balanced dataset category quantity detection result from data input through object selection improved from the original 80.36 to 90.26 . The advantages of this technology are real-time detection and statistical benefits. In addition to reduced labor costs, our intelligent detection reduces the probability of accidents.

Keywords YOLO, Vehicle detection, Intersections.

1. INTRODUCTION

In daily life, there are many hidden outdoor danger areas, such as traffic intersections. Accidents often occur in areas where traffic flow is complex. Therefore, it is important and indispensable to monitor those public places[1][2]. By using artificial intelligence (AI), objects can be automatically recognized from monitoring images. Then, after classifying moving objects at specific intersections, we can obtain instant results and analyze the immediate status of images, which further enhances safety. This technology is not only useful for the safety of outdoor areas, but also for campus security and other public places. The application of traffic monitoring in smart cities is a positive indicator. The purpose is to monitor and control the number of vehicles and pedestrians in certain areas. The frequency of accidents involving roads with excessive numbers of large vehicles and roads without traffic police control is relatively different from that of ordinary intersections. Therefore, it is more accurate to determine the type and flow rate of each vehicle at the intersection, making it possible to increase the visibility of control and intersection markings as a result, thus reducing the accident rate.

Recently, because deep learning and AI, both based on convolutional neural networks (CNN), have flourished, the accuracy of object detection and classification has greatly increased. Providing a large volume of relevant image data to the network model can enable automatic extraction of training features, and we can find regularity in the data. Deep learning has a multi-level structure, which is key to independent learning of various feature classifications. Deep learning has been used widely for vehicle classification[3][4][5]. Thus, we use it to classify vehicle objects from intersection images. By first establishing a deep learning module, we input the intersection images to the back-end, classify the moving objects, and record the number. Then, we achieve traffic control. This technique can also be applied to smart campuses to help identify the different needs of students and faculty.

During our research, we discovered that a good network model requires a good dataset to avoid unnecessary image training data in the model during training. Therefore, in order to establish a good dataset we need a well-designed model rule to ensure that all training images are necessary. From this, we establish an appropriate dataset and reduce misjudgment during testing. And apply actual intersection images, shown in Fig. 1

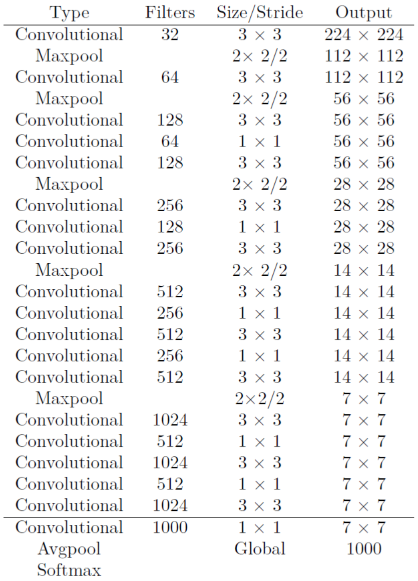
Fig. 1: Sample Image of Street Surveillance Video.

2. RELATED WORK

Among the many common network models, the faster R-CNN[6], the single-shot multi-box detector (SSD)-500[7], and the “you only look once” (YOLO) v2[8] are common fast network architectures applicable to our task. After analysis comparison is shown in Table 1. SSD500 and YOLOv2 mean average precision (mAP) are also 76.8, but YOLOv2’s 67 fps higher than SSD500’s 48 fps.

Table 1: Comparison of current common deep learning architectures mAP values and FPS

Model	mAP	FPS
Faster R-CNN[6]	76.4	5
SSD500[7]	76.8	19
YOLOv2[8]	76.8	**67**



Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2× 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2× 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2× 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2× 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2×2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

Fig. 2: YOLOv2 Network Architecture Diagram[8]

YOLO is also widely used in various places[9][10][11], and by many organizations for vehicle detection[12][13]. Based on our analysis and YOLO’s extensive use in vehicle classification, YOLO was selected for this experiment. Fig. 2 displays YOLO’s network architecture diagram.

3. METHODOLOGY

At present, the common dataset contains many categories. Our purpose is to identify objects from a monitored intersection. Therefore, we use real intersection images as training data and subdivide them into four categories of intersection imagery objects: large vehicles; small vehicles; motorcycles; and bicycles (as shown in Fig. 3). We build dataset rules to improve the quality of training results. By using YOLOv2 to evaluate this dataset after verifying accuracy, speed, etc., we verify the training results. Moreover, we use the same parameters and the same number of iterations (i.e., 15,000) to train.

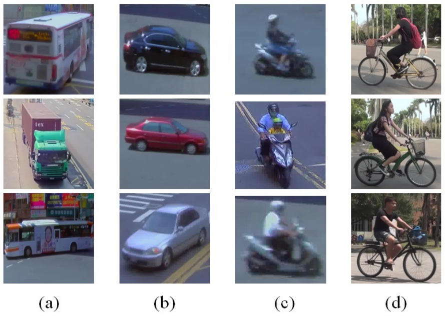


Fig. 3: (a) Large vehicles, (b) Small vehicles, (c) Motorcycles, (d) Bicycles.

3.1. Filtering all categories of small objects

Because tiny objects are located farther away from the intersection, there are very few that need to be identified. Moreover, they contain too few features, which may result in poor training results. Therefore, we assume that all datasets will remove tiny objects, as shown in Fig. 4, to improve the recognition rate.

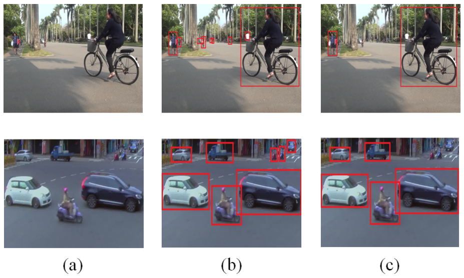


Fig. 4: (a) original picture, (b)filtering before, (c) filtering after.

Common CNN training using non-marked objects are classified as negative samples (i.e., background). However, if the removal have characteristic objects will result have characteristic positive samples become negative sample, it will affect training. We can avoid this situation by defining appropriate rules and filters. We must avoid false negative detection for too small object by defining appropriate rule for filtering out these objects. In this study, we only count those objects with object width (pixels) is larger or equal to set up threshold, otherwise we just discard it.

3.2. Balance the quantity each of category

During general training, we directly divide all annotation sets into training and verification data. All training data are used, and if it the datasets are self-created, there may be issues with uneven quantities in each category. Because of this, it is usually ineffective to train the entire dataset. With many categories, corrections are more frequent. Therefore, we design another experiment, set up the thresholds, and average all the training data (see Fig. 5).

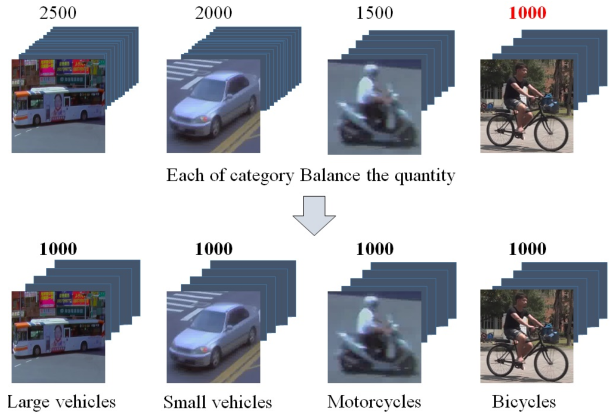


Fig. 5: Balance the quantity of Each category.

4. EXPERIMENTAL RESULTS

4.1. Filtering all categories of small objects

To define rules for properly removing small objects, we used self-creating datasets and set several different threshold, to test (Threshold=50, Threshold=100, Threshold=150), and we filtered each classification before and after the experiment. According to Table 2.

Table 2: Number of categories before and after filtering(A: Large vehicles, B: Small vehicles, C: Motorcycles, D: Bicycles)

Threshold	A	B	C	D
0	5487	4421	4345	1370
50	5355	3445	3698	1331
100	4090	2001	1344	849
150	2917	917	649	459

Using YOLO was effective for training and verifying data after filtering and for selecting suitable thresholds from it. According to Table 3, when threshold as 100, we achieved the threshold value of the dataset. It can also be proven that filtering too-small objects can improve the recognition rate, and the filtering threshold setting will adjust for different datasets and different needs.

Table 3: Each categories recognition rate compare. (A: Large vehicles, B: Small vehicles, C: Motorcycles, D: Bicycles)

Threshold	A	B	C	D	mAP
50	89.22	88.07	69.70	80.31	81.83
100	89.70	83.88	88.48	90.42	**88.12**
150	89.52	75.15	87.85	89.96	85.62

4.2. Balance the quantity each of category

Because of the different times and locations, there were many differences in the number of categories, and the amount of training in each category was imbalanced for self-created datasets. This led to poor training for smaller numbers of categories. Thus, we used the minimum number of categories as a balancing benchmark. As shown in Fig. 6, blue denotes the amount of imbalanced data, and orange denotes the amount of data after balancing.

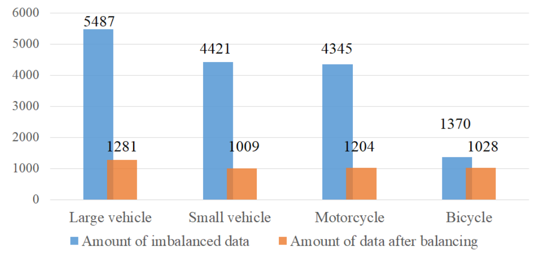


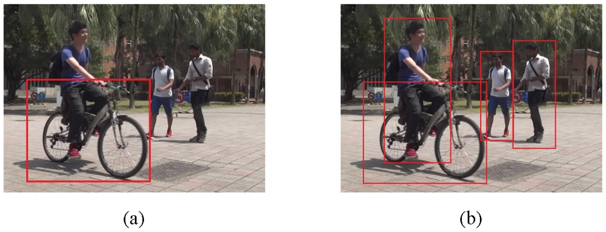
Fig. 6: The result of balance category object counting.

We used YOLO to train and verify balanced data, as shown in Table 4. After balancing the number of categories, some categories changed, but the overall recognition rate improved. Thus, it can be proven that balancing the amount of unbalanced data in each category helps improve training results.

Table 4: Comparison of results before and after balance. (A: Large vehicles, B: Small vehicles, C: Motorcycles, D: Bicycles)

	A	B	C	D	mAP
Before balance	88.66	84.92	67.17	80.67	80.36
After balance	88.65	86.77	61.27	88.49	**81.41**

Pedestrians in the intersection images were also an important category that we added. Furthermore, motorcycles and bicycles in the intersection images were marked as pedestrians, as shown in Fig. 7. Thus, the pedestrian category cannot be balanced.


(a)

(b)

Fig. 7: Add pedestrian marked data.

Pedestrian annotation data is shown in Fig. 8. Training results are shown in Table 5; the mAP is slightly decreased. It can be seen that this will affect the balancing of categories. If the category is deemed unnecessary, it may not be added to the training.

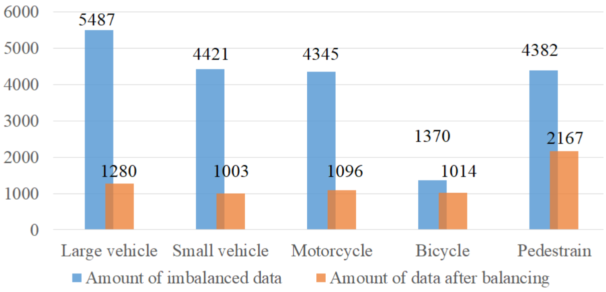


Fig. 8: The result of balance category object counting(Add pedestrian)

Table 5: The training results after adding pedestrian. (A: Large vehicles, B: Small vehicles, C: Motorcycles, D: Bicycles, E: Pedestrian)

	A	B	C	D	E	mAP
Balance	88.65	86.77	61.27	88.49		**81.41**
Added the “E” category + Balance	88.17	79.71	61.38	81.13	68.36	75.75

4.3. result comparison

Both the above experiments showed an effective increase in the recognition rate. Finally, the two experiments were merged, as shown in the Table 6. The input data are filtered by size, and the quantity is balanced, thereby showing an improvement from the original mAP 80.36 to an mAP 90.26.

Table 6: The experiment of result.(A: Filtering tiny object, B: Balance the number of each category)

Experiment	mAP
Undisposed	80.36
A	88.12
B	81.41
A + B	90.26

5. CONCLUSION

Self-created datasets often conceal many problems. The quality of training materials has a large influence on training results. Improving self-created datasets thus becomes an issue. This paper proposed two methods to improve the training data. One method was filtering out small objects and finding an appropriate threshold for screening. Another was balancing the quantity of data in each category to achieve balanced training times. The mAP was increased from the original mAP 80.36 to an mAP 90.26 through the size-screening and quantity-balancing methods.

ACKNOWLEDGEMENT

This research is supported by the Ministry of Science and Technology. The grant numbers are “106-2622-8-009-017”

REFERENCES

[1] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan, “Vision-based

occlusion handling and vehicle classification for traffic surveillance systems," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 2, pp. 80–92, 2018.

[2] Sitapa Rujikietgumjorn and Nattachai Watcharapinchai, "Vehicle detection with sub-class training using r-cnn for the ua-detrac benchmark," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–5.

[3] Kuan-Chung Wang, Yoga Dwi Pranata, and Jia-Ching Wang, "Automatic vehicle classification using center strengthened convolutional neural network," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2017*. IEEE, 2017, pp. 1075–1078.

[4] Wei Liu, Miaohui Zhang, Zhiming Luo, and Yuanzheng Cai, "An ensemble deep learning method for vehicle type classification on visual traffic surveillance sensors," *IEEE Access*, vol. 5, pp. 24417–24425, 2017.

[5] Hongbo Gao, Bo Cheng, Jianqiang Wang, Keqiang Li, Jianhui Zhao, and Deyi Li, "Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, 2018.

[6] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[8] Joseph Redmon and Ali Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017.

[9] Hyeok-June Jeong, Kyeong-Sik Park, and Young-Guk Ha, "Image preprocessing for efficient training of yolo deep learning networks," in *Big Data and Smart Computing (BigComp), 2018 IEEE International Conference on*. IEEE, 2018, pp. 635–637.

[10] VV Molchanov, BV Vishnyakov, YV Vizilter, OV Vishnyakova, and VA Knyaz, "Pedestrian detection in video surveillance using fully convolutional yolo neural network," in *Automated Visual Inspection and Machine Vision II*. International Society for Optics and Photonics, 2017, vol. 10334, p. 103340Q.

[11] PR Peiming Ren, WF Wei Fang, and SD Djahel, "A novel yolo-based real-time people counting approach," 2017.

[12] Jing Tao, Hongbo Wang, Xinyu Zhang, Xiaoyu Li, and Huawei Yang, "An object detection system based on yolo in traffic scene," .

[13] MH Putra, ZM Yussof, KC Lim, and SI Salim, "Convolutional neural network for person and car detection using yolo framework," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1-7, pp. 67–71, 2018.