

A YOLOv3-Based Smart City Application for Children's Playgrounds

Mehmet Fatih İNKAYA* , Hakan GÜRKAN 

Department of Electrical and Electronics Engineering, Bursa Technical University, 16310 Bursa, TURKEY

Abstract

According to the reports of Public Health Institution, approximately 250,000 rabies-risk animal bites occur per year in Turkey. Most of these bites are caused by dogs and most of the victims are the children who play in playgrounds. With the development of deep learning-based computer vision technology, autonomous detection of dangerous objects (handguns, knives, dogs, etc.) in these children's playgrounds has become a crucial security application. In this paper, a real-time dog detection model based on YOLOv3 deep learning algorithm is proposed as a new smart city security application and this model is applied to the selected children's playground. Firstly, in view of the problem of insufficient stray dog image data in the original datasets, new images of stray dogs have been taken from an animal shelter and they have been added to the dataset. These new images have effectively enriched the diversity of training data and improved the training performance of the proposed model. The proposed model has been optimized by utilizing various hyperparameters and the results have been compared with each other. The model with the best evaluation scores is proposed and applied to detect dogs automatically by the fast emergency station located in the selected children's playground. The real-time application has achieved 82.59% of AP with adjusted hyperparameters.

Keywords: YOLOv3; Smart City; Children's Playground; Dog Detection; Fast Emergency Station.

Cite this paper as:

İNKAYA, M.F., GÜRKAN, H. (2021). A YOLOv3-based Smart City Application for Children's Playgrounds. Journal of Innovative Science and Engineering. 5(1): 25-40

*Corresponding author: Mehmet Fatih İNKAYA
E-mail: fatihinkaya@hotmail.com

Received Date: 20/10/2020
Accepted Date: 13/12/2020
© Copyright 2021 by
Bursa Technical University. Available
online at <http://jise.btu.edu.tr/>



The works published in Journal of Innovative Science and Engineering (JISE) are licensed under a Creative Commons Attribution-NonCommercial 4.0 International License.

1. Introduction

The world's population living in cities is rising day by day and it is estimated to reach 70% of the entire population in 2050 [1]. Although the rapid growth of urban population causes many problems for inhabitants of the city, it is a major function of the municipalities to find solutions to these problems. Bursa Metropolitan Municipality launched an application called "fast emergency stations" to make children's playgrounds safe and healthy in March 2019.

The fast emergency station, which is connected to the public security center through the internet, consists of an IP video intercom system and a camera module [2]. Firstly, the person who feels in danger presses the button and starts the video call. Then, the security personnel answer the call and get the information. After that, if the personnel validate the event by the camera, they perform the security procedure. All the video calls made with the fast emergency station between March 2019 and September 2019 were recorded. When the records were examined retrospectively, two important issues were identified. First issue was that most of the children did not use the fast emergency station properly. They preferred to escape instead of pressing the button in case of danger and they pressed the button like a game when there was no threat. The second issue was that unvaccinated stray dogs were reported to be the major danger in the playgrounds. In this study, the fast emergency station was enhanced to detect stray dogs not manually but automatically to fix these issues.

The children are the main victims of dog bites because they are at the eye level of the dogs and this is recognized as a threat by the dogs. Due to the children's shortness, dogs bite on their head or neck and this bite can be fatal. According to the reports of Centers for Disease Control and Prevention Institute, each year approximately 4.5 million dog bites occur in the USA. 471 people died between 2005 and 2018 because of dog bites [3]. According to Public Health Institution, 283,185 rabies-risk animal bites occurred in 2018 in Turkey [4]. In Ankara 79.1% of total 25,480 rabies-risk animal bites between 2005 and 2009 were caused by dogs [5]. In Eskisehir 71.6% of rabies-risk animal bites were caused by dogs and most of the bites were reported to occur in the summer months when the children were on holiday. 48.2% of these dogs were stray dogs [6]. In Yildirim province of Bursa, 70.7% of total rabies-risk animal bites between 2011 and 2012 were caused by dogs. 75.9% of them were stray dogs. 50.1% of the victims were between age of 0-19 and 39.8% of the bites were reported to happen in summer months [7]. At the emergency medicine clinic of Istanbul Haydarpaşa Numune Training and Research Hospital, it was observed that 35.2% of total 10,974 rabies-risk animal bites were caused by dogs. 80.8% of the dogs were stray dogs. 33.7% of the victims were between age of 0-19 and 29.8% of the bites were reported to occur in summer months [8]. Also, 107 zoonotic infections including 37 bacterial, 13 fungal, 29 viral, 28 parasitic (3 trematodes, 7 cestodes, 10 nematodes and 8 protozoan) have been reported from Turkey [9].

Until Alexnet [10] won the ImageNet competition in 2012, object detection had been carried out using traditional methods. Starting from this date, more successful results have been obtained in object detection by using deep learning models.

Recently, in the smart security field, deep learning-based object left/removed [11], smoke detection [12], face recognition [13], abnormal event detection [14] etc. video analytics applications have become key applications to prevent false alarms and to reduce the need for human verification.

In dangerous object detection, Lai et al. used a Tensorflow-based implementation of the Overfeat network to detect weapons and achieved 93% test accuracy with 0.77 fps, which is too slow for any real time detection [15]. Olmos et al. used Faster R-CNN with VGG-16-based classifier to detect handgun and achieved 84.21% of AP with 5 fps speed [16]. Fang et al. used Tinier-YOLO to detect dogs and achieved the result 67.9 % of AP with 25 fps speed [17].

Deep learning-based object detection algorithms have two types: "region-proposal" and "single shot". The region-proposal networks (R-CNN, Fast R-CNN, Faster R-CNN, R-FCN etc.) first determine the areas likely to be found and then run a classifier in these regions. Although it produces more accurate estimates, this type of algorithm is slower because it processes 2 rows on the image. However, single shot networks (YOLO, SSD, RetinaNet etc.) regress the image at once, estimate the class of objects and their position at the same time. Therefore, they are faster than region-proposal networks and more suitable for real time detection.

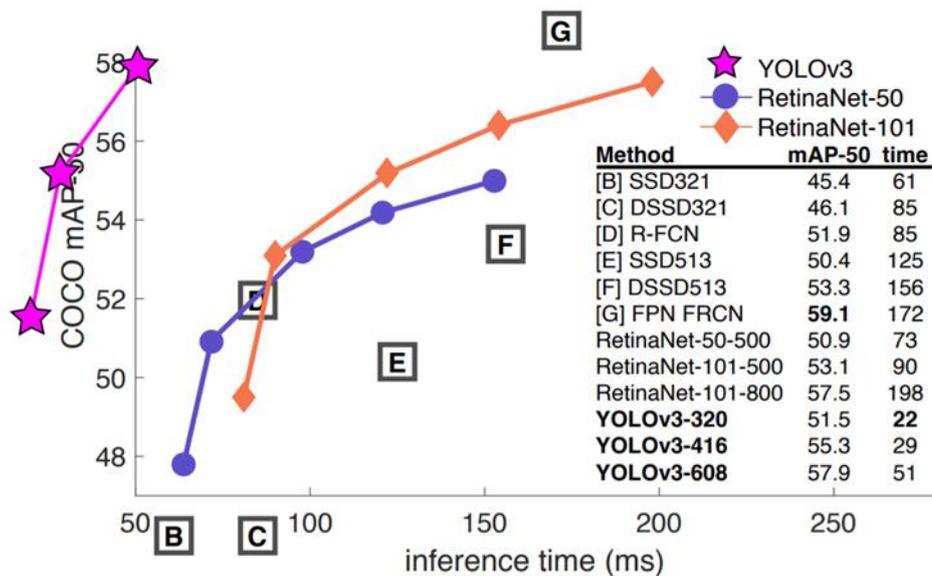


Figure 1. Performance of mean average precision (mAP) and time on the COCO dataset [18]

YOLOv3 [18] achieves good results on the mAP50 metric with a faster runtime speed comparing with other models as shown in Figure 1. Hence, in this study a YOLOv3 model has been preferred for object detection application. Nevertheless, YOLOv3 model has not been widely applied in dangerous object detection in children’s playgrounds.

In this study, object detection test was performed first by pre-trained YOLOv3 model, which was automatically obtained with the Darknet setup. However, it was observed that the model was unsuccessful to detect stray dogs as shown in the Figure 2.



Figure 2. Predictions of pre-trained YOLOv3 model

It was considered that these false predictions were caused by the 80 classes and insufficient stray dog image data in the original COCO dataset [19].

Later, a new dataset that contained two classes and 878 images was created by adding 241 images of stray dogs from an animal shelter, 236 images of children from the playground and 401 images of both dogs and humans from the Open Images V6 dataset [20]. Training sessions with various hyperparameters on this new training dataset were held to optimize the model and the results were compared with each other. The model with the best evaluation scores was proposed and applied to detect dogs automatically by the fast emergency station located in the selected children's playground. The real-time application achieved 82.59% of AP with adjusted hyperparameters.

2. Material and Methods

The fast emergency station, which is located at the children's playground in Hudavendigar Park of Bursa, is shown in the Figure 3.

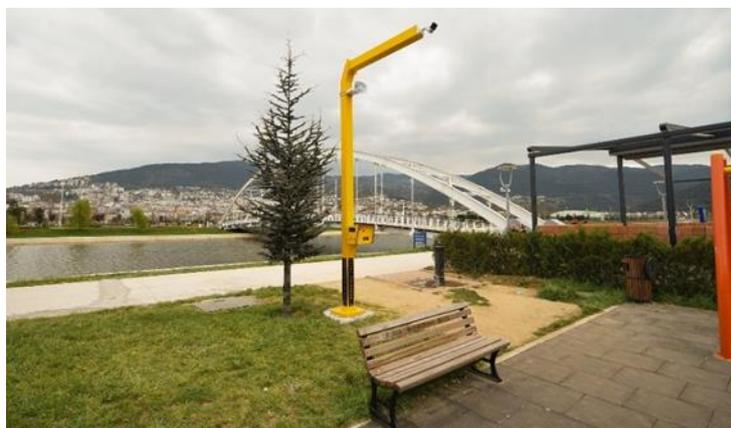


Figure 3: Fast emergency station in Hudavendigar Park [2]

In this study, the fast emergency station was enhanced to detect stray dogs not manually but automatically by using a YOLOv3 algorithm.

2.1. YOLOv3 Algoritihm

YOLOv3 (You Only Look Once version 3) algorithm is one of the single shot object detection algorithms. Unlike region proposal algorithms, YOLOv3 applies single neural network to an image, which makes it much faster.

YOLOv3 network consists of two main parts: Darknet-53 feature extractor and YOLO detector. Darknet-53 is a convolutional network which is pre-trained on ImageNet. It has 53 consecutive (1x1 followed by 3x3) convolutional layers, which are followed by residual layers, as shown in Figure 4.

	Type	Filters	Size	Output
	Image			416x416
	Convolutional	32	3x3 / 1	416x416
	Convolutional	64	3x3 / 2	208x208
	Convolutional	32	1x1 / 1	208x208
1x	Convolutional	64	3x3 / 1	208x208
	Residual			208x208
	Convolutional	128	3x3 / 2	104x104
	Convolutional	64	1x1 / 1	104x104
2x	Convolutional	128	3x3 / 1	104x104
	Residual			104x104
	Convolutional	256	3x3 / 2	52x52
	Convolutional	128	1x1 / 1	52x52
8x	Convolutional	256	3x3 / 1	52x52
	Residual			52x52
	Convolutional	512	3x3 / 2	26x26
	Convolutional	256	1x1 / 1	26x26
8x	Convolutional	512	3x3 / 1	26x26
	Residual			26x26
	Convolutional	1024	3x3 / 2	13x13
	Convolutional	512	1x1 / 1	13x13
4x	Convolutional	1024	3x3 / 1	13x13
	Residual			13x13
	Avgpool			
	Connected			
	Softmax			

Figure 4: Darknet-53 architecture [18]

In order to use Darknet-53 network as a feature extractor average pooling, connected layers and softmax activation layers are removed as seen in Figure 5. Since YOLOv3 network is a multi-scaled detector, it needs features from multiple scales. Thus, the last 3 residual blocks are connected with detector layers, assuming the input image size 416x416, three scale vectors will be 13x13, 26x26 and 52x52.

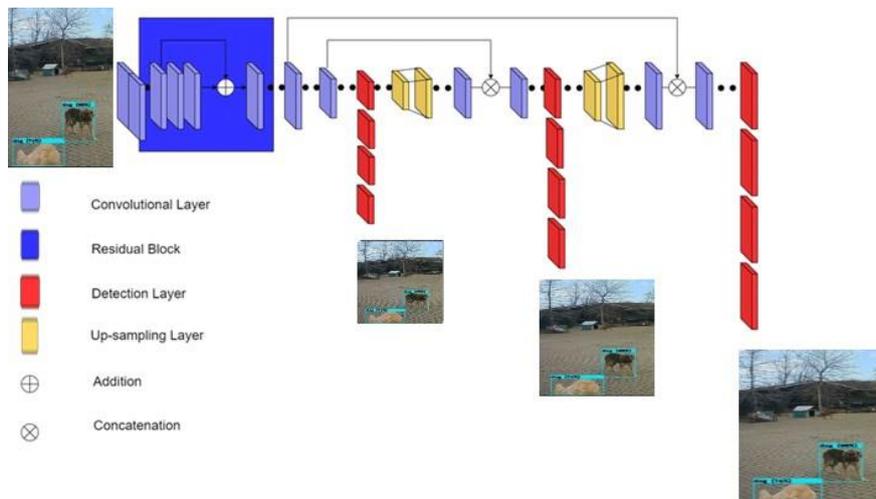


Figure 5: YOLOv3 architecture [21]

The image is divided into 3 different scale grids (13 x 13, 26 x 26, 52 x 52) to detect different size of objects.

At 3 different scales, each grid cell can predict 3 boxes using 3 anchors to detect multiple objects. Anchor boxes are defined after training datasets with different scales and aspect ratios to detect specific classes. For example, in this study one of the anchor boxes is a vertical rectangle like the shape of a person and the other one is a horizontal rectangle like the shape of a dog.

YOLOv3 uses k-means clustering on the training set to find good priors (anchor boxes) instead of choosing them by hand. The network predicts the width (t_w) and height (t_h) of the box as offsets from cluster centroids and predicts center coordinates (t_x, t_y) of the box relative to the location of sigmoid function. The calculation of absolute location and size of a predicted box by an anchor box is shown in the Figure 6.

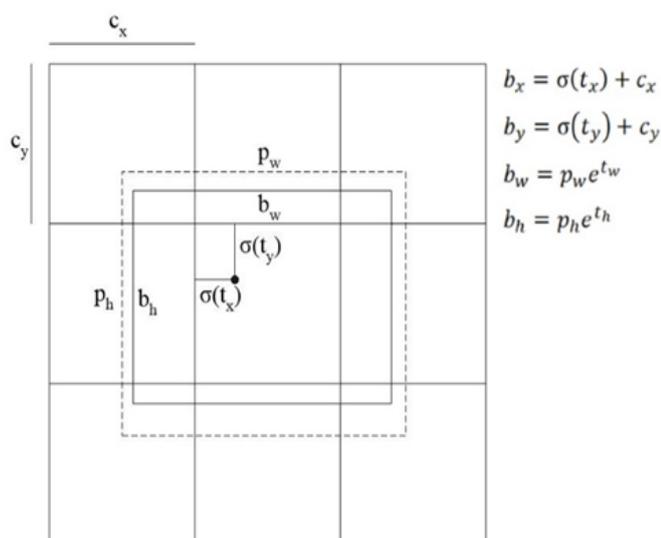


Figure 6: Prediction of a box by an anchor [18]

$\sigma(t_x)$ and $\sigma(t_y)$ are the centroid locations relative to the grid cell which are normalized with sigmoid function between 0 and 1. The centroid of predicted box (b_x, b_y) is calculated by adding the absolute location of the top left corner of the grid cell values (c_x, c_y) to these $\sigma(t_x)$ and $\sigma(t_y)$ values. The size of predicted box (b_w, b_h) is calculated with the values p_w and p_h which show the predefined size of the anchor. Considering the probability that t_w and t_h values are negative, e^{t_w} and e^{t_h} are taken to make it positive.

The network also predicts an objectness (confidence) score and class probabilities for each of the anchor boxes. Objectness score indicates if predicted box contains an object while class probabilities show which class this object in the box belongs to. The class probabilities for all classes are calculated individually. Since the original YOLOv3 network uses the COCO dataset, which is a large-scale dataset with object annotations containing 80 classes, 328,000 images with 2,500,000 instances, a total of 80 probabilities are predicted. Thus, the original YOLOv3 network predicts $(4 + 1 + 80 = 85)$ values for each anchor box and $(3 \times 85 = 255)$ values for each grid cell.

Since the network divides the image into 3 different scale grids, the network predicts $((52 \times 52) + (26 \times 26) + (13 \times 13)) \times 3 = 10,647$ anchor boxes in total and most of them will have a very low objectness score. To keep the best boxes and eliminate the others, YOLOv3 uses non-maximal suppression (NMS) algorithm. The first step of NMS is to remove all these boxes with lower objectness score below the threshold. The second step is to select the best box with the highest objectness score and eliminate all the other boxes whose intersection over union (IoU) value is above a specific IoU threshold with this best box. Afterwards, NMS the best predictions are made as shown in Figure 7.

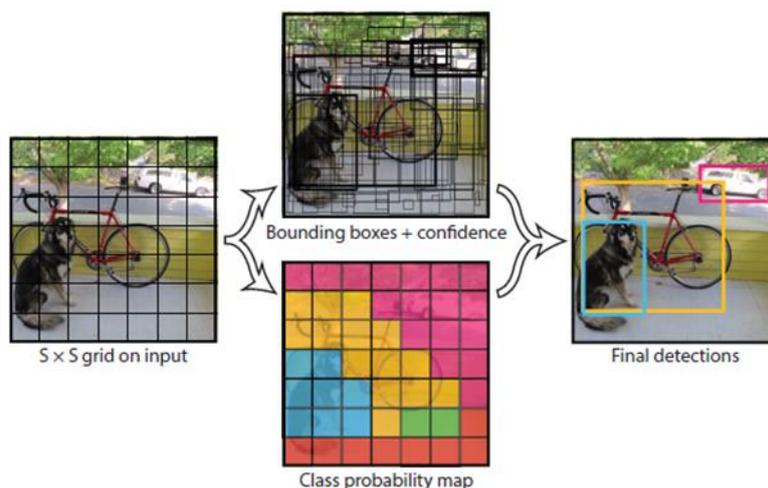


Figure 7: Non-Maximal Suppression [22]

2.2. Preparing Computer Environment

In this study, a computer with Intel Core i7, 16 GB RAM and NVIDIA GeForce GTX 1050Ti was used to train the model. Darknet framework, OpenCV v.3.3.0, CUDA 10.0 and CuDNN 7.4.1 were compiled to run on GPU computing for a fast training. Convolutional weights from the Darknet53 model (darknet53.conv.74) that had been pre-trained on Imagenet were used as initial weights for training in Darknet framework.

2.3. Preparing New Image Dataset

To optimize the neural network for children’s playground, firstly the recordings of fast emergency station in Hudavendigar Park at different times were downloaded from the NVR (network video recorder) which is located in the public security center. The recordings were divided into frames and 236 of these frames containing the children with different scenes were selected to add into a new dataset. Some images taken with the fast emergency station in the children’s playground are presented in Figure 8.



Figure 8: Images taken with the fast emergency station in the children’s playground.

Second, the fast emergency station was taken to Bursa Bademli Sevgi animal shelter to take stray dog videos. The videos were divided into images and 241 of these images containing the stray dogs with different scenes were selected to add into the new dataset. Some images taken with the fast emergency station in the animal shelter are presented in Figure 9.



Figure 9: Images taken with the fast emergency station in the children’s playground.

In order to enrich the dataset, 401 images including both humans and dogs were taken from the Open Images Dataset V6, which is the largest open source dataset with object annotations containing 600 classes, 9,000,000 images with 16,000,000 bounding boxes. Some of these 401 images taken from the Open Images Dataset V6 are presented in Figure 10.



Figure 10: Images from open images dataset v6 [20]

The images in the new dataset were labeled in YOLO format by using LabelImg [23] open source graphical image annotation tool as shown in Figure 11.



Figure 11: Images from open images dataset v6 [20]

3. Experimental Results and Discussions

We trained the model by adjusting the hyperparameters (batch, subdivisions, width, height) to detect objects most accurately. In this way, 6 different training sessions were carried out on the new dataset. TP, FP and FN values of each training are shown in Table 1.

Table 1. TP, FP and FN values in trainings with different hyperparameters

	WidthxHeight	Batch	Subdivisions	TP	FP	FN
Training 1	320x320	64	64	247	54	194
Training 2	320x320	64	32	291	65	150
Training 3	320x320	32	32	220	47	221
Training 4	320x320	32	16	294	73	147
Training 5	416x416	64	64	263	78	178
Training 6	416x416	32	32	247	79	194

Higher input image size (608x608) and batch-subdivision combinations (64-16, 32-8) of the hyperparameters in Table 1 were tried and the hyperparameters had to be updated because CUDA memory error was received by the existing hardware.

The average prediction number of the trainings was 507 where the ground truth was 441. The lowest Type-1 error was made in Training-3 while the lowest Type-2 error was made in Training-4. There is a trade-off between precision and recall. Although networks with high recall and low precision produce a large number of predictions, many of the predictions may be inaccurate. Networks with high precision and low recall produce fewer but relatively more accurate predictions.

Table 2 indicates that Training-3, which produces the least number of predictions (488), has the highest precision value (0.82). However, the production of a small number of predictions brought Type-2 error with it, thus causing the recall value to be low (0.50). As a result, Training-3 ranked last in terms of F1-score (0.62).

Although, compared to other trainings, the precision value of Training-4 (0.80) is relatively low, it has the highest recall (0.67), which made this training reach the highest F1-score (0.73). Training-2, which produces the closest predictions (506) to the average number of predictions, has the highest precision (0,82) and the 2nd highest recall value (0,66). This balanced situation enabled Training-2 to reach the highest F1-score (0.73).

The highest IoU value was obtained in Training-2 and the lowest IoU value in Training-6.

On the condition that batch and subdivisions parameters are constant, the average loss decreases with the increase of the input image size. The average loss decreases with increasing batch and subdivisions values on the condition that the image size is constant. Precision, recall, F1-score, average IoU and average loss values of each training are shown in Table 2.

Table 2. Evaluation metrics of the trainings.

	Precision	Recall	F1-Score	IoU (%)	Average Loss
Training-1	0,82	0,56	0,67	61,42	0,4034
Training-2	0,82	0,66	0,73	62,07	0,4330
Training-3	0,82	0,50	0,62	60,60	0,4979
Training-4	0,80	0,67	0,73	60,89	0,5703
Training-5	0,77	0,60	0,67	56,89	0,3391
Training-6	0,76	0,56	0,64	54,58	0,4943

The best precision-recall points of the trainings are shown in Figure 12.

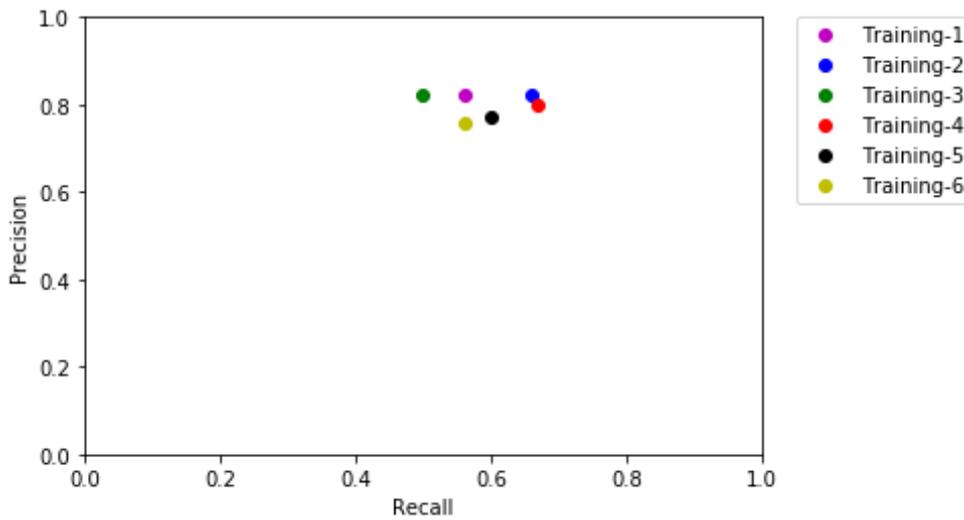


Figure 12: The best precision-recall points of the trainings

Training-3 (green dot) is the farthest from the upper right corner compared to the others, which indicates the lowest F1-score obtained in this training.

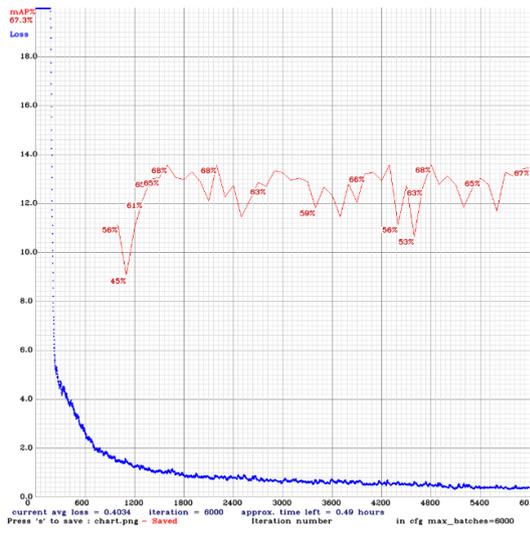
Training-2 and Training-4 (blue and red dots, respectively) are closer to the upper right corner compared to the others, which indicates the highest F1-scores obtained in these two trainings.

AP (average precision) of each class and mAP (mean average precision) of each model are shown in Table 3.

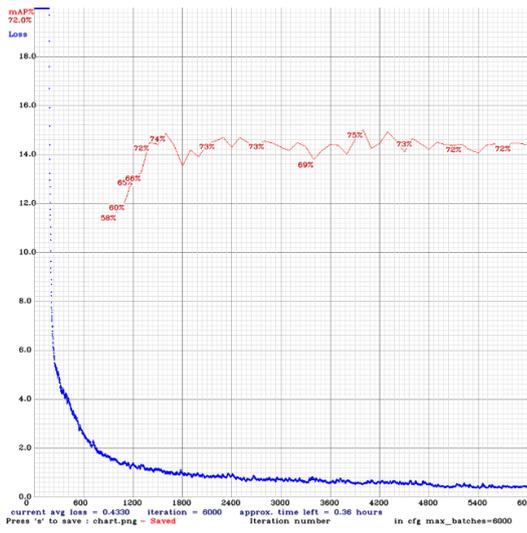
Table 3. APs and mAPs.

	AP_Dog	AP_Person	mAP (%)
Training-1	75,45	60,38	67,91
Training-2	82,59	67,52	75,05
Training-3	78,20	60,17	69,19
Training-4	83,06	65,51	74,28
Training-5	81,88	61,66	71,77
Training-6	81,03	60,83	70,93

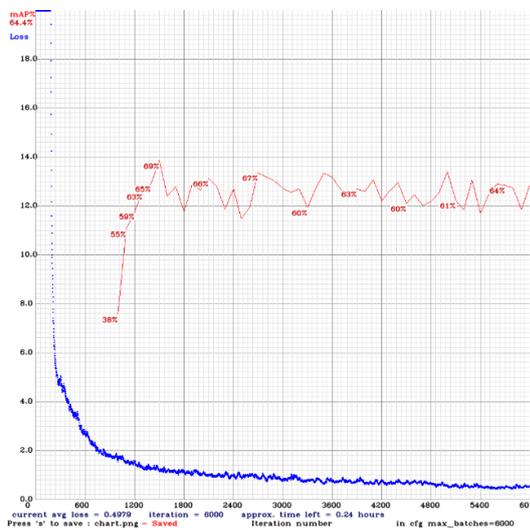
Changes in mAP and average loss values related to the number of iterations for all trainings are shown in Figure 13.



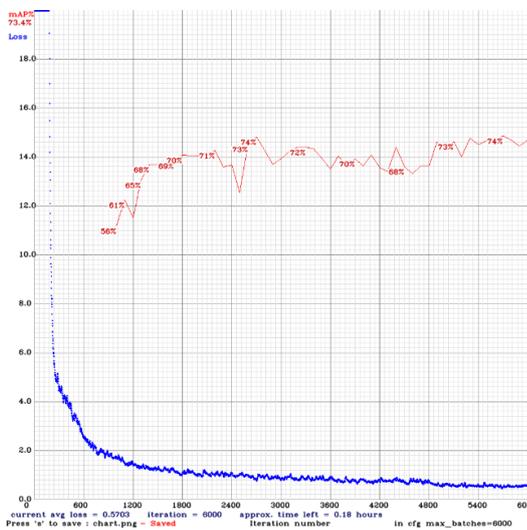
(a)



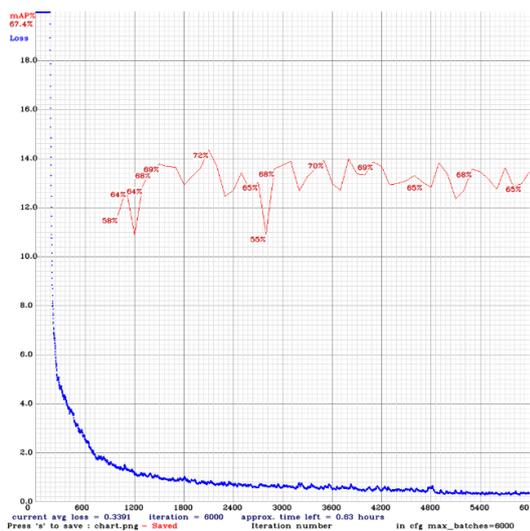
(b)



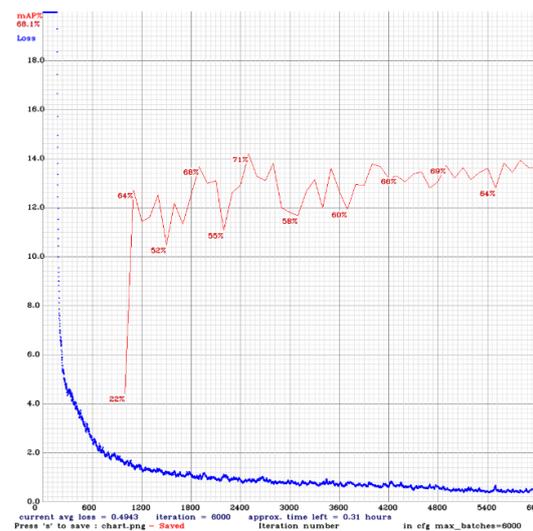
(c)



(d)



(e)



(f)

Figure 13: Average loss and mAP chart of (a) Training-1 (b) Training-2 (c) Training-3 (d) Training-4 (e) Training-5 (f) Training-6

By comparing the charts and the figures above, it was determined that the network was trained most successfully in Training-2, considering the highest F1-score, IoU and mAP value. The neural network weight which was obtained in Training-2 was selected as the best weight of the proposed YOLOv3 network.

The object detection tests were performed in the children's playground and animal shelter as seen in Fig.14 and 15. The validation tests showed that the proposed model outperforms the pre-trained YOLOv3 model which was automatically obtained with the Darknet setup in terms of detection accuracy.

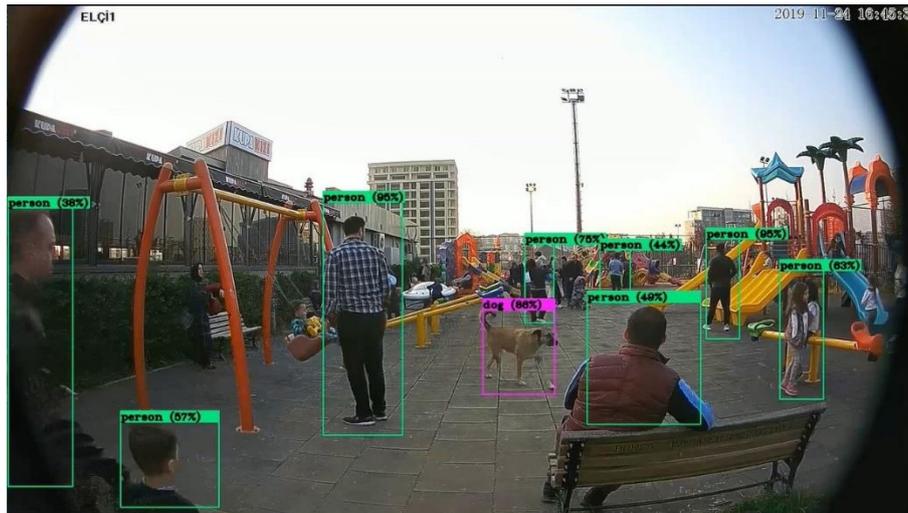


Figure 14: Validation test in the children's playground

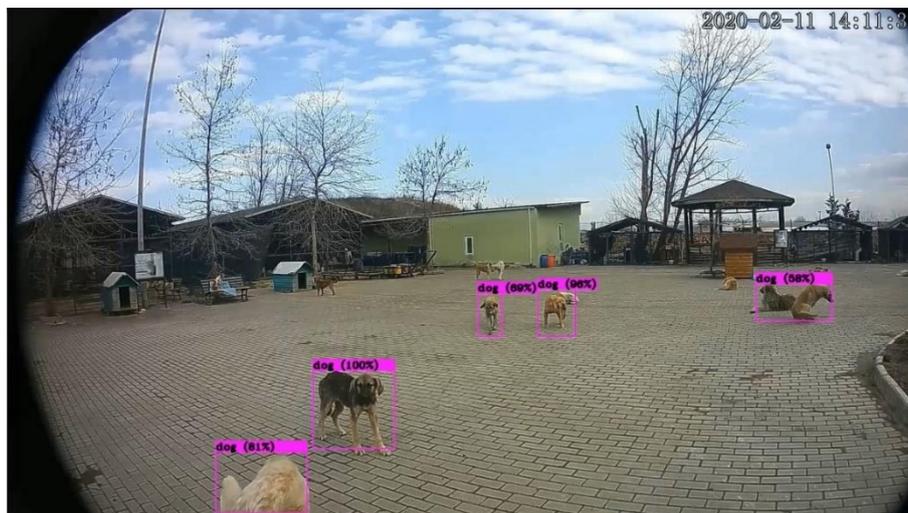


Figure 15: Validation test in the animal shelter

Moreover, the validation tests were performed on different resolutions (320x320 and 608x608) of the same video. Thirty scenes that contained 65 dogs were selected from the video. While the model successfully detected 52 dogs at 25 fps in these scenes with the resolution of 320x320, it successfully detected 60 dogs at 10.2 fps in these scenes with the resolution of 608x608. In this way, it was possible to make a trade-off between accuracy and speed by changing the resolution of the camera on the fast emergency station.

The performance of the proposed model is compared with similar studies in Table 4. It should be emphasized that the dataset, the resolution and the target object used in these studies are completely different. Nevertheless, the results given in this table might provide a comparison between these models. As seen from the results shown in Table 4, the performance of the proposed model is superior to the other models in terms of AP, especially for 25fps. Besides, the proposed model provides a good balance of speed and accuracy that the application needed.

Table 4. Comparison of Similar Studies.

Research paper	Detection type	Algorithm	DataSet	AP	Fps
Developing a Real-Time Gun Detection Classifier [15]	Handgun	Overfeat	Imagenet	93%	0.77
Automatic Handgun Detection Alarm in Videos Using Deep Learning [16]	Handgun	Faster R-CNN with VGG-16 Based Classifier	Imagenet	84.21%	5
Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments [17]	Dog	Tinier YOLO	PASCAL VOC	67.9%	25.1
The Proposed Model	Dog	YOLOv3	Customized Dataset	82.59%	25

4. Conclusion

In this paper, we proposed a real-time dog detection model for children's playgrounds based on the YOLOv3 deep learning algorithm as a new smart city security application. Firstly, we obtained new images of stray dogs from an animal shelter to overcome the problem of insufficient data. These new images effectively enriched the diversity of training data and improved the performance of the proposed model. The model with the best evaluation scores was proposed and applied to detect dogs automatically by the fast emergency station located in the selected children's playground. The real-time application achieved 82.59% of AP with adjusted hyperparameters. Moreover, the validation tests demonstrated that the proposed model outperforms the pre-trained YOLOv3 model automatically obtained with the Darknet setup in terms of the detection accuracy.

Unfortunately, due to the Covid-19 pandemic situation, the children's playground was closed for a certain period. Also, very few children visited these playgrounds after the children's playgrounds were opened.

In future studies, the fast emergency station will be enhanced to detect other dangerous objects such as handguns and knives in order to prevent violence against children.

Acknowledgement

We would like to thank Bursa Metropolitan Municipality for providing us with the fast emergency stations and the other resources to conduct our experiments.

References

- [1] <https://data.worldbank.org/indicator/SP.URB.TOTL.IN.ZS?contextual=default> Accessed: 10 August 2020.
- [2] <https://www.bursa.bel.tr/cocuklara-buyuksehir-korumasi/haber/27347> Accessed: 10 August 2020.
- [3] <https://www.dogsbite.org/reports/14-year-dog-bite-fatality-chart-trends-in-3-periods-2005-2018.pdf> Accessed: 08 October 2020.
- [4] Aylan, O., Baykam, N., Güner, R., Kara, A., Köksal, İ., Seçer, M., Tülek, N., Ünal, N. (2019). T.C. Sağlık Bakanlığı Kuduz Profilaksi Rehberi.
- [5] Aksoy, M., Demirbaş, B., Maden, F., Şimşek, Ç., Özlü, A., Kaya, M., Çulha, G., Topaç, O., Arı, E., Yılmaz, H., Gürbüz, F. (2010). Ankara İlinde 2005–2009 yılları arasında görülen şüpheli ısırıkların ve kuduz aşılmasının değerlendirilmesi, Türkiye 3. EKMUD Kongresi Özet Kitabı, p.199.
- [6] Gunduz, T., Elcioglu, O. and Balci, Y. (2011). An Evaluation of Dog And Cat Bites For a Five Year Period: A Sample Case From Eskisehir, *Turkish Journal of Trauma and Emergency Surgery*, 17(2):133–140.
- [7] Karadağ, M., Çatak, B., Baştürk, S. and Elmas, Ş. (2014). Assessment of Rabies-Risk Contact Notifications In Yıldırım, District of Bursa, *Turkish Journal of Family Practice*, 18(3): 117–121.
- [8] Tekyol, D. (2019). Evaluation of rabies risky contact cases admitted to emergency department last year, *Haydarpaşa Numune Medical Journal*, 2019.83723.
- [9] İnci, A., Doğanay, M., Özdarendeli, A., Düzlü, Ö. and Yıldırım, A. (2018). Overview of Zoonotic Diseases in Turkey: The One Health Concept and Future Threats, *Türkiye Parazitoloji Dergisi*, 42(1):39–80.
- [10] Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017). ImageNet Classification With Deep Convolutional Neural Networks, *Commun. ACM*, 60(6): 84–90.
- [11] Luna, E., San Miguel, J., Ortego, D. and Martínez, J. (2018). Abandoned Object Detection in Video-Surveillance: Survey and Comparison, *Sensors*, 18(12): 4290.
- [12] Namozov, A. and Cho, Y. I. (2018). An Efficient Deep Learning Algorithm for Fire and Smoke Detection with Limited Data, *Advances in Electrical and Computer Engineering*, 18: 121–128.
- [13] Sun, Y., Liang, D., Wang, X. and Tang, X. (2015). DeepID3: Face Recognition with Very Deep Neural Networks, *arXiv:1502.00873*.
- [14] Wang, T., Miao, Z., Chen, Y., Zhou, Y., Shan, G. and Snoussi, H. (2019). AED-Net: An Abnormal Event Detection Network, *Engineering*, 5(5): 930–939.
- [15] Lai, J. and Maples, S. (2017). Developing a Real-Time Gun Detection Classifier, *World Academy of Science, Stanford University*, p.4.
- [16] Olmos, R., Tabik, S. and Herrera, F. (2017). Automatic Handgun Detection Alarm in Videos Using Deep Learning, *Neurocomputing*, 275: 66-72.
- [17] Fang, W., Wang, L. and Ren, P. (2020). Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments, *IEEE Access*, pp. 1-1.
- [18] Redmon, J. and Farhadi, A. (2018). YOLOv3: An Incremental Improvement, *arXiv:1804.02767*
- [19] <https://cocodataset.org/> Accessed: 30 November 2020.
- [20] <https://storage.googleapis.com/openimages/web/index.html/> Accessed: 30 November 2020.

- [21] <https://developer.nvidia.com/blog/deploying-a-scalable-object-detection-inference-pipeline/> Accessed 27 September 2020.
- [22] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, Jun. 2016, pp.779–788.
- [23] <https://github.com/tzutalin/labelImg/> Accessed: 30 November 2020.