Tiny-YOLO object detection supplemented with geometrical data

1st Ivan Khokhlov
Wave Processes and Control Systems Lab, MIPT
Moscow, Russia
khohklov.iyu@gmail.com

2nd Egor Davydenko
Wave Processes and Control Systems Lab, MIPT
Moscow, Russia
egordv@gmail.com

3rd Ilya Osokin
Wave Processes and Control Systems Lab, MIPT
Moscow, Russia
kefir8888@gmail.com

4th Ilya Ryakin
Wave Processes and Control Systems Lab, MIPT
Moscow, Russia
ryakin.is@phystech.edu

5th Azer Babaev
Wave Processes and Control Systems Lab, MIPT
Moscow, Russia
7684067@mail.ru

6th Vladimir Litvinenko
Wave Processes and Control Systems Lab, MIPT
Moscow, Russia
litvinenko.vv@phystech.edu

7th Roman Gorbachev
Wave Processes and Control Systems Lab, MIPT
Moscow, Russia

Abstract—**We propose a method of improving detection precision (mAP) with the help of the prior knowledge about the scene geometry: we assume the scene to be a plane with objects placed on it. We focus our attention on autonomous robots, so given the robot's dimensions and the inclination angles of the camera, it is possible to predict the spatial scale for each pixel of the input frame. With slightly modified *YOLOv3-tiny* we demonstrate that the detection supplemented by the scale channel, further referred as S, outperforms standard RGB-based detection with small computational overhead.**

Index Terms—object detection, neural network, data fusion

I. INTRODUCTION

A. Research Area Overview

Our interest is mainly focused on autonomous bipedal robots. We validate our algorithms and ideas by taking part in world competition - RoboCup. The competition is a soccer game between two teams of autonomously playing robots. Each team consists of four robots maximum. Robots are supposed be similar to the humans in sensors, body structure, proportions and center of mass position. They can communicate via Wi-Fi network with each other and the referee. The playing area looks like FIFA football field, but is much smaller - 6×9 meters.

The robots' ability to play increases significantly each year. Our robots can localize and move pretty good and stable. The next huge step to improve the level of the game is to implement opponent robots detection. Also, it is important to increase quality and distance of ball detection. These two points push us to investigate methods of object detection in our specific conditions.

B. Problem Statement

We have mono camera FLIR Blackfly GigE[1] with wide-angle lens and Intel NUC Core i5[2] as computational platform [1]. We need to detect opponent robots and ball on the playing field. Obstacle detection is an important part of navigation stack of humanoid soccer robot. It allows robot to behave much more like human player and not to hit other robots during the game. For this goal we can call obstacle any object on the field that cannot be bypassed by the robot. This usually includes opponent robots, ball and human referee. This can be reformulated to an obstacle is any part of the playing field which is not green, and its not a field marking line. This formalization of an obstacle can be turned into a simple classic computer vision pipeline that is capable of running in real time on the robot for obstacle detection as described below. The second approach to this problem is object detection using neural networks. This will give us more information about the current field state. We can obtain information about number of opponent robots and their coordinates. The application of this approach is more promising because it will give us new data to improve strategy. It is important that we are strongly limited by CPU computational abilities in neural network-related tasks.

C. Related work

In the [2] paper authors work in similar setting, but they obtain the geometrical data from the video stream. In this work a static camera is used, which is not the case in the autonomous robots competitions.

[3] paper presents an approach basing on rectifying the image before detection. It allows the neural network to not to learn features for the distorted image. For the problem of detecting distant objects captured by a camera of the robot such a rectification is not possible due to the heavy distortions arising.

The approach in [4] relies on the joint analysis of RGB and Depth images. Contours and region proposals are used to build depth encoding, and after the feature extraction fused

---

[1]https://www.flir.com/products/blackfly-gige
[2]https://www.intel.ru/content/www/ru/ru/products/boards-kits/nuc/kits/nuc7i5bnh.html

with the RGB features. This approach cannot be adopted to the considered problem since the real Depth data is not present.

II. Distance transform-based approach

Since the obstacle is formalized as part of the playing field that is not green, it is possible to directly discriminate pixels of an input image to green and not green using so called green filter classifier that consists of RGB to HSV domain color transformation and thresholding. The resulting image of this filter is shown in the Fig. 4.

The sample input image contains different types of not green objects on the field: two opponent robots (black and white), human referee, and field marking lines. All of these objects are correctly classified as not green by the green filter. The resultant image is further processed by morphological operations (erosion and dilation) [5] to filter out noisy pixels that appear as fake green on non-green objects.

The next step is to distinguish marking lines among the other objects. For filtering the field markings we use the prior knowledge of the width of a marking line that is constant in any part of the field. The assumed line marking width in pixels can be calculated for all input image pixels using the data about camera orientation relatively to the field planar surface available in real-time from robots internal model subsystem. To filter objects that are thinner than the field marking line we use the distance transform approach [5]. The result of the distance transform applied to the black and white image is a single-channel image in which the value of the pixel is equal to the distance to the closest non-zero pixel in source image. The result of the distance transform of a sample green filter output can be seen in Fig. 3(left). By thresholding this image with the help of pixel-wise knowledge of field line marking width we come to the result shown in Fig. 3(right). The available for walking areas are shown in green color.

The last part of the filtering is the boundary detection of the non-zero clusters found on the thresholded distance transform result. This boundary is shown by yellow lines in Fig. IV. Each pixel of this boundary is then checked by the obstacle filter, and if it is close enough to the robot, it is treated as an obstacle and then processed in robots navigation stack.

III. Deep Learning Approach

A. Dataset, model, augmentation

For this approach we have used $YOLOv3 - tiny$ DNN [6] as a baseline for the object detection. The choice was dictated by the conditions the net is deployed in and the properties of the architecture itself, meaning only CPU (no graphic accelerator on board), relatively small available dataset, small number of classes and required real-time or almost real-time inference.

We have collected and marked a real-world dataset of 1705 frames from the camera of the robot (see section Problem Formulation for camera specification), considering the following classes: ball (standard Robocup ball), goal post, rhoban (the type of robots that we use).

We have used the following augmentation methods: vertical and horizontal translation (10%), rotation (5 degrees), vertical and horizontal shear (2 degrees), scale (10%), horizontal reflection, HSV Saturation and Value change (50%). The augmentation was applied during training.

B. Geometrical data incorporation

We have attempted to exploit the specific properties of the problem, particularly the knowledge about the scale of every pixel in the scene. Knowing the inclination angles of the camera, the spatial dimensions of the robot and the calibration matrix of the camera it is possible to obtain the scale on the field for every pixel. This scale data that is equivalent to the depth map can be present in the form of a single channel image of the same dimensions as the original.

Since the scene is assumed to be a plane, it is possible to interpolate **S** between pixels with the stride of 32 without loss of quality, which makes the computational overhead for calculating **S** very small. Experiments show that inferring 4-channel net on 1088×1440 images is approximately 1.232 times more time-consuming than 3-channel.

The intuition behind this approach is that $YOLO$ can probably generalize the connection between the size of the object and the class of the object and improve the quality of the predictions.

C. Current results, discussion

TABLE I
COMPARISON WITH DIFFERENT IoU BETWEEN YOLOv3-TINY RGB AND RGB-S FOR **MAP** METRIC

	YOLOv3-tiny	
	RGB	**RGB-S**
mAP@0.5	0.84	0.872
mAP@0.6	0.819	0.867
mAP@0.7	0.766	0.818
mAP@0.8	0.638	0.734
mAP@0.9	0.503	0.599

D. Future work

There are many approaches and hypothesis to be tried out in the chosen direction of adding geometrical data directly to the input data, among them we can highlight the following:

- Try regular (not tiny) YOLO, Spiking YOLO [7] and another architectures.
- Try less straightforward adding of the scale data, for instance add few independent convolutional layers before fusing it with the color data.

IV. Conclusion

The robot's behaviour strongly relies on the precise object detection, since the distances to the objects are calculated out of their coordinates in the frame. Table I shows that the proposed method significantly outperforms baseline with high demands to the detection precision, see the **mAP@0.9** line.

The quality improvement is achieved with relatively small computational overhead, which is critically important for the autonomous robots due to the limited resources. The method's application does not require any extra data apart from **S**, calculated from the robot's parameters and measurements that were already present. The incorporation of the geometrical data was made on the setup with a single RGB camera.

The proposed approach is not limited to the autonomous robotics applications, it can be used to improve detection accuracy when the scene geometry is known.

REFERENCES

[1] S. Semendyaev, D. Zavrazhnov, I. Khokhlov, P. Senichkin, N. Koperskii, I. Ryakin, and V. Litvinenko, "Starkit team–team description paper," 2019.

[2] D. Xu, W. Xie, and A. Zisserman, "Geometry-aware video object detection for static cameras," *arXiv preprint arXiv:1909.03140*, 2019.

[3] J. Cai, H. Chen, L. Kneip, and S. Schwertfeger, "Improving cnn-based planar object detection with geometric prior knowledge," *arXiv preprint arXiv:1909.10245*, 2019.

[4] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *European conference on computer vision*. Springer, 2014, pp. 345–360.

[5] R. Fisher, S. Perkins, A. Walker, and E. Wolfart, "Hypermedia image processing reference (hipr)," *Artificial Intelligence - AI*, 01 1996.

[6] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv*, 2018.

[7] S. J. Kim, S. Park, B. Na, and S. Yoon, "Spiking-yolo: Spiking neural network for real-time object detection," *CoRR*, vol. abs/1903.06530, 2019. [Online]. Available: http://arxiv.org/abs/1903.06530
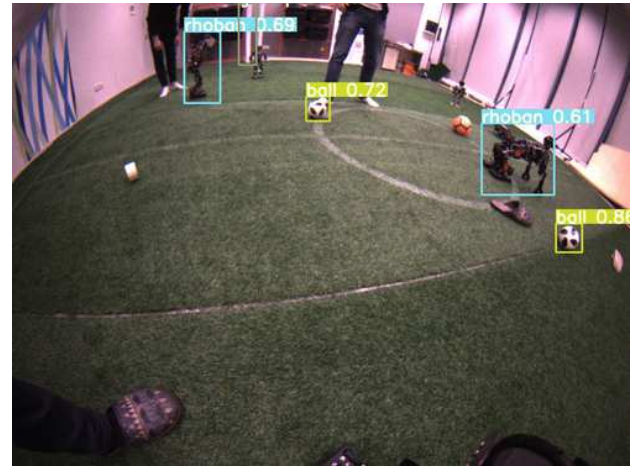
Fig. 1. Left image - RGB, right - RGB-**S**, note that rhoban in the right side of the field is missing in the first of them. Also **S** channel improves the bbox accuracy, see left rhoban occurrence.

Fig. 2. Left image - RGB, right - RGB-**S**. These images demonstrate improvement not only in the detection precision, but also in classification: in the left the foot of Nao v.6 robot is confused with the ball, while the rhoban in the right side of the frame is missing.
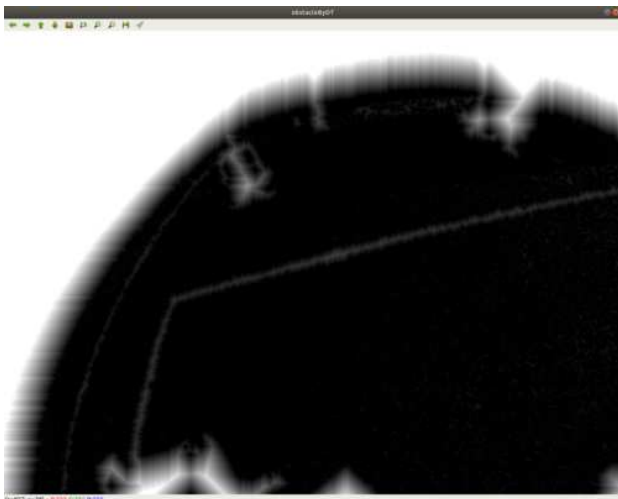

Fig. 3. Distance transform output (left) and its thresholding result using knowledge of assumed field marking width at any source image pixel (right). Green colored area is walkable or too far.
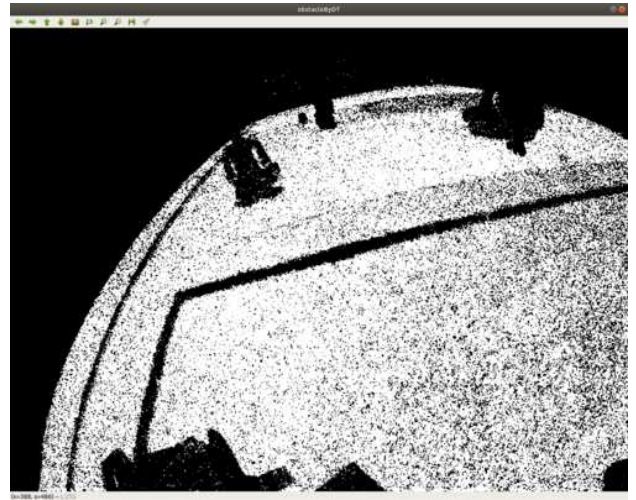
Fig. 4. Input frame from robots camera and the output of green filter.
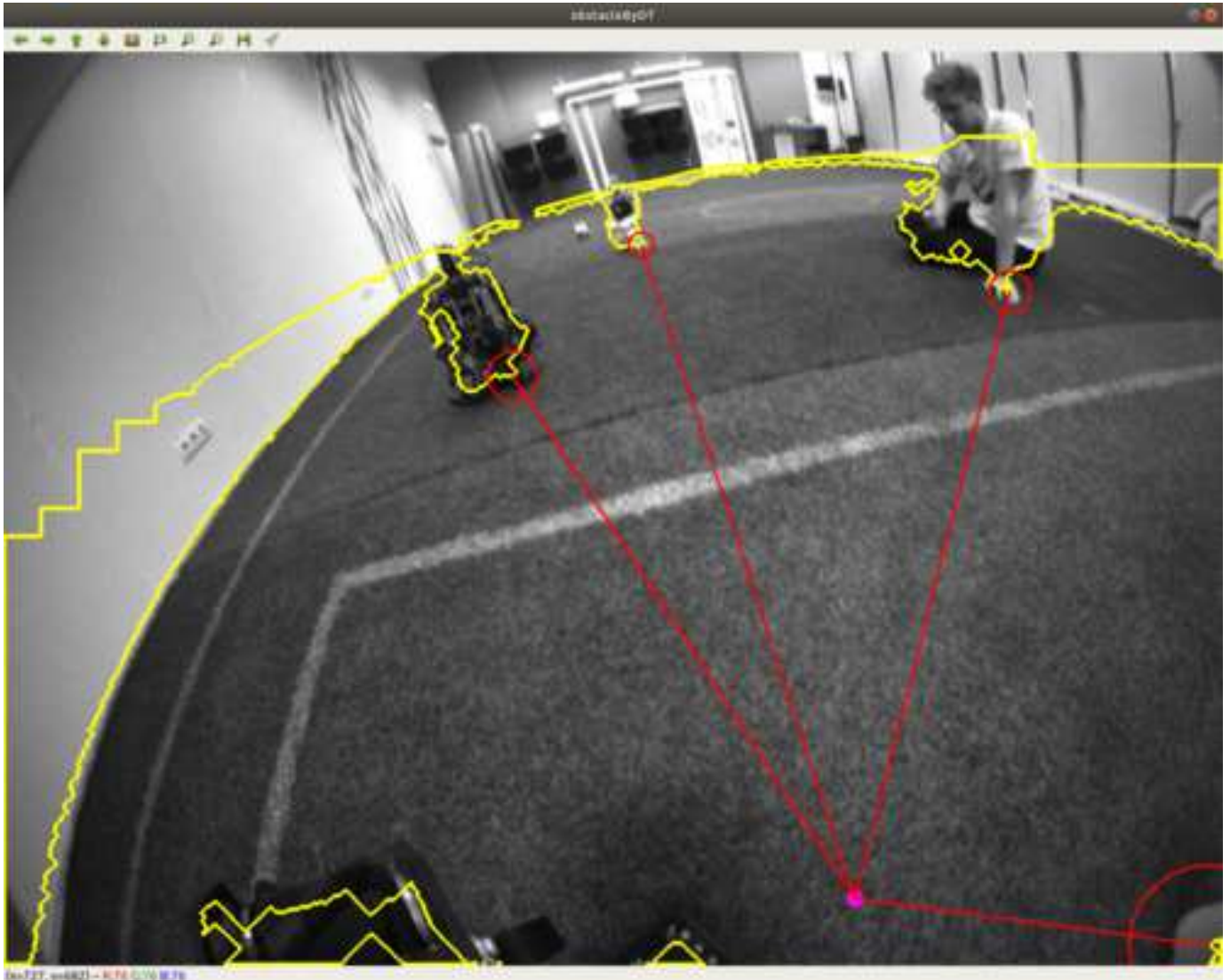

Fig. 5. Distance transform clusters boundary marked by yellow. Boundary pixels classified as a part of an obstacle marked by red lines pointing from the robots next step position on the field marked by magenta dot.