A Learning Framework Towards Real-time Detection and Localization of a Ball for Robotic Table Tennis System

Yongsheng Zhao[1], Jun Wu[1], Yifeng Zhu[1], Hongxiang Yu[1], and Rong Xiong[1]

Abstract— **As a real-time serving system interacting with a highly dynamic environment, robotic table tennis system has a high requirement against the accuracy and robustness of real-time detection and localization of a ping-pong ball. Relative to its size, the ball is a high speed flying-spinning object. The existing methods use general features such as color and shape to detect and localize the ball, which rigidly depends on the prior knowledge. Their performance is susceptible to the change of the environment, *e.g.*, the light condition, the color of ball, and the disturbance of human players' presence in the image. In this paper, we propose a learning framework that trains a convolutional neural network to detect and localize a ball with high accuracy. It learns useful features from data directly without any prior knowledge. Therefore, the proposed method can effectively deal with the situation when the ball's color is changing in real-time. And it is more robust to the light condition and the disturbance of human players' presence. The effectiveness and accuracy of the method is verified using the collected data set, in comparison with the state-of-the-art method.**

I. INTRODUCTION

Robotic table tennis system is an excellent research platform for real-time sensing, intelligent decision making, and servo motion planning. With the purpose to promote the development and application of robotic technologies, John Billingsley [1] first proposed robot table tennis competition in 1983. Many robotic table tennis systems [2]–[17] have been designed and developed. Several systems achieve human-level performance that can play against human players in a long rally at all kinds of speed, such as the humanoid robot ”Wu & Kong” designed by Zhejiang University [13] and the parallel link robot ”FORPHEUS” designed by Omron [17].

Considering the size of table (standard size $2.74 \times 1.525 \times 0.76m$), table tennis is definitely one of the fastest games in the world. For a ball with flying speed of $10m/s$, it takes about $0.274s$ to fly from one side to another. Therefore, robotic table tennis system highly requires the capability of quick-reaction from the real-time perception to motion planning and control. Moreover, the direction of flying velocity is various in every round and the spin effect would result in a large trajectory bias. In order to successfully return a ball back to the desired landing point, the robot has to estimate the ball's motion state and predict its trajectory accurately, which depends on the precision of localization to a large extend. In addition, the temporal resolution of real-time perception also plays an important role in the motion state estimation and trajectory prediction. The higher the frame rate of a camera is, the more accurate the motion state estimation will be. As far as we know, the detection and localization of a ping-pong ball with high accuracy and temporal resolution is the main challenge for a robotic table tennis system in real-time perception of the environment.

Currently, the most widely used ping-pong ball is in color of pure white or yellow, which is a distinguishable featue in the context of green or blue table for detection and localization. The projection of a ping-pong ball on the image always is a circle with a limited size from any point of view. Consequently, most of the existing methods [6], [7], [11]–[14], [18], [19] used the ball's features of color, shape and size for detection and localization. A general pipeline of these methods is 1) Pre-process the image using adjacent frame different (or with collected background template) to detect the areas that contain moving objects, such as human player, ping-pong ball and paddle, and further binarize the processed image using a intensity threshold that is manually selected. 2) Transform the image from RGB color space to HSV color space and select the areas that contain a ping-pong ball according to the rule of thumb min-max thresholds in H and S Channel. 3) Based on the processed image, search the ball's contours using man-made rules and select the contour with highest confidence as the detected ball. 4) Compute the centroid of the ball according to the shape feature.

In most cases, those methods mentioned above can detect and localize the ping-pong ball with high accuracy in real-time, which helps the robot to percept the ball and play with human players in continuous rally. However, the performance of these method is susceptible to the light condition, *i.e.*, when the luminance or the white balance changes, the performance would get worse. Especially when the ball's color changes, these methods mostly fail to detect the ball. Since 2014, Chinese Table Tennis Association has introduced a two-color toned ball in the matches of China Super league. Half of the surface of ball is yellow and the other half is white, which makes the spin velocity more visible and helps the audience have a better understanding of the game. Obviously, these previous methods could not work effectively under this condition.

Compared to traditional computer vision algorithms, Convolutional Neural Network (CNN) [20], [21] has proven to be a powerful and efficient tool for hierarchichal feature extraction, especially in deep layers, which has outperformed traditional methods in the competition of Large Scale Visual

*This work was supported by the National Nature Science Foundation of China (Grant No.61473258 and U1609210)

[1] Yongsheng Zhao, Jun Wu, Yifeng Zhu, Hongxiang Yu, and Rong Xiong are with the State Key Laboratory of Industrial Control and Technology, Zhejiang University, Hangzhou, 310027, P. R. China. Rong Xiong is the corresponding author. `rxiong@iipc.zju.edu.cn`

Recogniton Challenge [22]–[25]. Threrfore, the well pre-trained deep CNNs [22]–[24] have a huge application in object detection, image classification, and segmentation.

In 2014, Ross Girshick [26]–[28] proposed a Region-based Convolutional Neural Network (R-CNN) that achieved a state-of-the-art performance on object detection and semantic segmentation. It uses a well pre-trained deep CNN to extract useful features from region proposals and then simultaneously regress the region proposals with bounding box and classifies objects in region proposals.

Inspired by the idea of R-CNN, we proposed a learning framework that can detect and localize the ping-pong ball in real-time with high accuracy and robustness. First of all, convolutional neural network is used to automatically learn the features of ping-pong ball in different colors by classifying the images of background and ping-pong ball. The pre-trained convolutional neural network can effectively detect the ball using the learned features. Second, a self-learned spatial softmax layer (the fixed spatial softmax layer was proposed by Levine [24]) accompanied by a ReLU layer was proposed to localize the ball by computing the average centroid of the detected feature pixels. Thirdly, two fully connected layers are further connected to the spatial softmax layer to regress the ball's location on the image by fine-tuning the convolutional layers. The neural network is trained using a large data set of pictures that contains ping-pong ball in different colors, which are collected under various light conditions.

The rest of the paper is organized as follows. Section II introduces the detailed mathematical formulation and architecture of the proposed learning framework. Section III is the collection and annotation of the data set of ping-pong balls in three different colors. In section IV, we present the training result and conduct comparison of experiments to verify the effectiveness and accuracy of the proposed method. In the end, section V summarizes the advantages and disadvantages of the proposed learning framework and gives a prospect on the future work.

II. Mathematical Formulation and Architecture of The Learning Framework

The goal of our work is to localize the ball's centroid in image coordinate accurately. In deep learning, it usually belongs to a regression task. However, we treat it as a binary segmentation task, *i.e.*, the ball is 1 and the background is 0, which achieves better performance.

In this paper, we propose a hierarchical neural network with a similar architecture to R-CNN. It has two branches that sharing the same convolutional layers, one is for the classification between ball and background and the other one is for the localization of the ball's centroid. The classification task contributes to the pre-training of convolutional layers, which can effectively extract the distinguishable feature of ping-pong ball and background. On the other hand, the localization task further finetune the pre-trained convolutional layer, which would accurately segment the ball from background.

The architecture of our learning framework is summarized in Fig. 1. It contains 2 convolutional layers accompanied by ReLU layer and max pooling layer. The first convolutional layer filters the $64 \times 48 \times 3$ input image with 16 kernels of size $3 \times 3 \times 3$ with a stride of 1 pixels. The second convolutional layer takes the output of the first convolutional layer (nonlinearized and pooled) as its input and filters the input with 4 kernels of size $3 \times 3 \times 16$ with a stride of 1 pixels. Then two fully connected layers are linked to the convolutional layers for classification, the first fully connected layer has 96 neurons and the second one has 2 neurons. As for localization, one self-learned spatial softmax layer is connected to the convolutional layers followed by two fully connected layers with 96 and 2 neurons. The self-learned spatial softmax layer and the fully connected layers will fine tune the convolutional layers that pre-trained by the classfication task, evaluate the distribution of the probability of each pixel belongs to ping-pong ball or background using the output feature map of convolutional layer, and localize the ball's centroid on the image accurately.

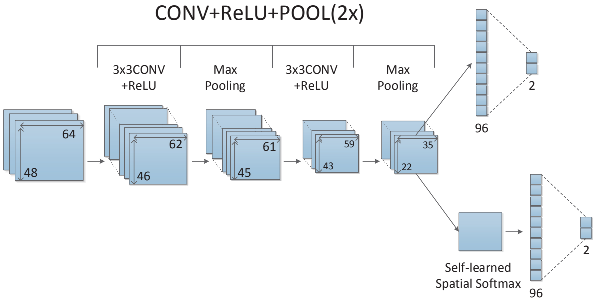


Fig. 1. An illustration of the Architecture of proposed Learning Framework, explicitly showing that the classification task and localization task share the same convolutional layers. The input to the network is a 64×48 sliding window on 640×480 image batches collected from cameras.

Usually, the more layers and the more neurons in each layer of the neural network, the better of the performance in hierarchical feature extraction. However, the computation time in forward step would increase with respect to the size of neural network. The architecture of the proposed learning framework is chosen by multi-trials, satisfying requirements of accuracy of feature extraction and high-speed computation.

A. *Improved Spatial Softmax Layer*

Using pure fully connected layers to regress the ball's centroid on image may work in this scenario, but it it not a good choice from the point of view of accuracy and efficiency. Here we propose an self-learned spatial softmax layer using a softmax layer and a full convolutional layer. It first compute the probability distribution of each pixel belongs to the ping-pong ball on the feature map and then compute the coordinates of the ball's centroid directly, which is more efficient and accurate.

The pipeline of the self-learned spatial softmax is shown in Fig. 2. Since the value of the ball's feature pixel is much larger than the value of the background's feature pixel on the input feature map, the softmax layer can obviously highlight the ball's feature and compute the probability of each pixel belongs to the ball. The following full convolutional layer filters the $30 \times 22 \times 4$ feature maps with 16 kernel of size $1 \times 1 \times 8$ with a stride of 1 pixels. Apparently, this convolutional layer would further make the ball's feature standout and suppress the background's feature by weightedly add the probability distribution maps.

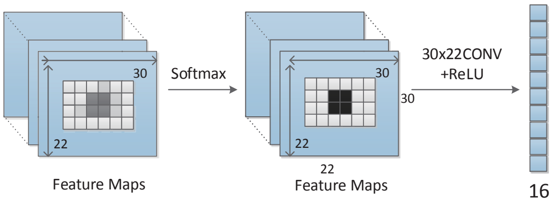


Fig. 2. An illustration of the improved spatial softmax layer. It contains one softmax layer, 16 $30 \times 22 \times 4$ full convolutional layer, one ReLU layer. The input to the spatial softmax layer is the 30×22 output featue maps of convolutional layers.

The spatial layer essentially is a full convolutional layer with kernel size the same as the size of input feature map. The convolution operation can be formulated as:

$$O = \sum_{u=1}^{U} \sum_{v=1}^{V} \omega(u,\ v) I(u,\ v) \tag{1}$$

where I denotes the input feature map, O denotes the output feature map, U denotes the width of input feature map, v denotes the height of input feature map, ω denotes the weights of the convolutional kernel. In this paper, $U = 30$ and $V = 22$.

For a general convolutional layer, the weights of each kernel is initialized randomly and further trained by back-propagation method. However,the fixed spatial layer [24] is a special full convolutional layer composed of 2 kernels with constant weights. As shown in equation (2) and (3), each row of weights in kernel v is initialized as its index and each column of weights in kernel u is initialized as its index.

$$\omega_v = \begin{bmatrix} 1 & 1 & \dots & 1 \\ 2 & 2 & \dots & 2 \\ \vdots & \vdots & \ddots & \vdots \\ 22 & 22 & \dots & 22 \end{bmatrix} \tag{2}$$

$$\omega_u = \begin{bmatrix} 1 & 2 & \dots & 30 \\ 1 & 2 & \dots & 30 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 2 & \dots & 30 \end{bmatrix} \tag{3}$$

According to (1), the spatial layer can compute the relative image coordinate of the detected object, given the well-trained binary feature map. Because the ball's feature is a circle, spatial layer can effectively compute the relative coordinate of its centroid. In this paper, we proposed a self-learned spatial softmax layer that the weights of full convolutional layer are initialized randomly just as a general convolutional layer and its weights can be well learned in the progress of localization. In the end, two fully connected layer with 96 and 2 neurons are used to fit the real coordinate.

III. Data Collection and Annotation

The performance of a convolutional neural network not only depends on its hierarchical architecture but also relies on the training data set to a large extend. The data set for both training and testing the proposed learning framework is collected and annotated by ourselves.

Fig. 3 shows the configuration of a binocular vision system used in the paper. The vision system consists of two Point Grey Research Grasshopper GRAS-03K2C CCD cameras. Its frame rate is 120 fps (up to 200 fps) and resolution is 640×480 pixels. The lens used in the vision system has $4mm$ focal length, which guarantees a broad field of view. The height of the vision system to the table is about $1.5m$ and it can observe the ping-pong ball in more than half of the table.

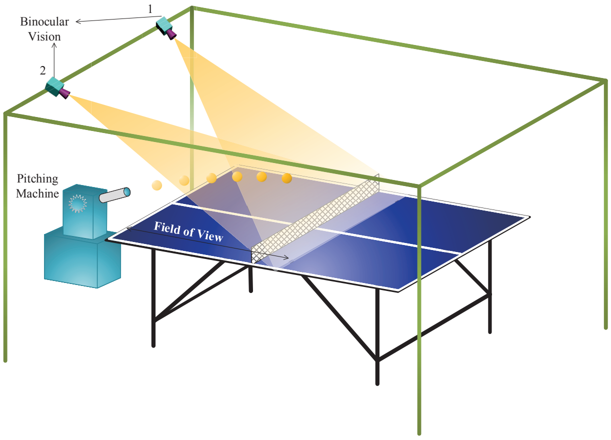
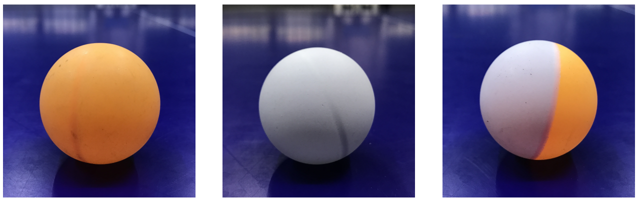


Fig. 3. Configuration of Binocular Vision System

As far as we know, the most popular ping-pong ball used currently are in yellow and white. In future, the two-color toned ball that makes the spin state more visible would be more and more frequently used. In order to make our robot adaptive to all three types of balls, as shown in Fig. 4, we collect a data set of them using our vision system.

In order to make sure the learning framework has a robust performance under various light conditions, we collected the data set by changing light conditions. As shown in Fig. 5, each row is the ball in one color under 5 light conditions. The first column is defined as normal light condition that used in our robotic table tennis system currently. The second column and third column are collected in a bright and dark light condition. And the fourth and fifth columns are collected in light with yellow and blue white balance.



(a) Yellow (b) White (c) Yellow-White

Fig. 4. Three kinds of ping-pong ball with different colors

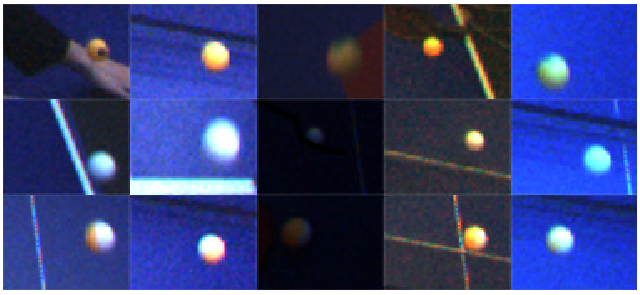


Fig. 5. The Samples of Collected Data Set

Totally, 2800 images with a resolution of 640 × 480 are collected. The ball's centroid on image is annotated manually using a circle to fit the ball's contour, as shown in Fig. 6. We crop the annotated images into smaller images with a resolution of 64 × 48, acting like a sliding window. 5000 cropped images are selected as a data set for classification and 28000 cropped images are selected as a data set for localization, which ensures the data diversity and avoids redundancy effectively.

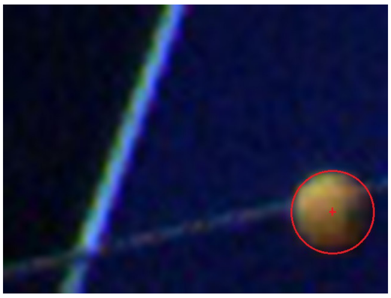


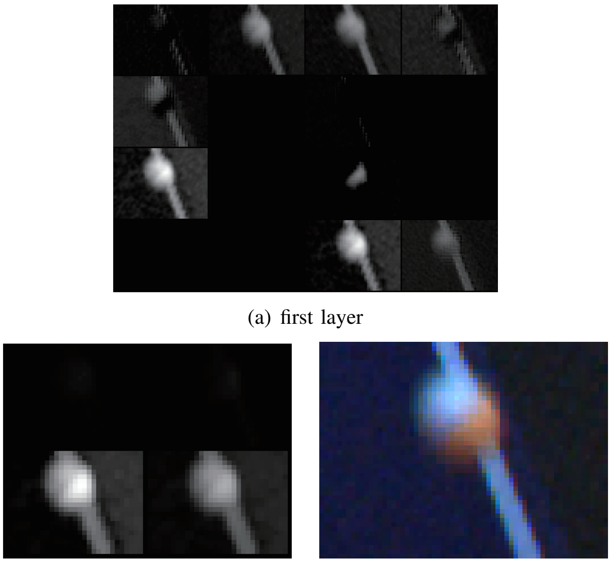
Fig. 6. Annotation of the ball's centroid

IV. Experiments and Results

Using the collected data set, we first train the proposed learning framework by a classification task as well as a segmentation and regression task. Then a comparison of experiments is conducted with a color-based method [13], which effectively verifies the accuracy and robustness of the proposed learning framework. The learning framework is built in caffe [29] and trained using a i7-2600k CPU.

A. Training Process

The convolutional layers are first trained by classifying the data set into ball and background. By doing this, the convolutional layers learn the ball's distinguishable features and can detect the ball from background effectively. Fig. 7 shows the output feature maps of the two convolutional layers in the proposed learning framework pre-trained by classification. Fig. 7(c) is the 64 × 48 input data of a two-color toned ball on the table's edge. As we can see, the convolutional layers detect the features of white table edge, the yellow semi-ball and white semi-ball, which are useful for classifying the ball and background.



(a) first layer

(b) second layer (c) input data

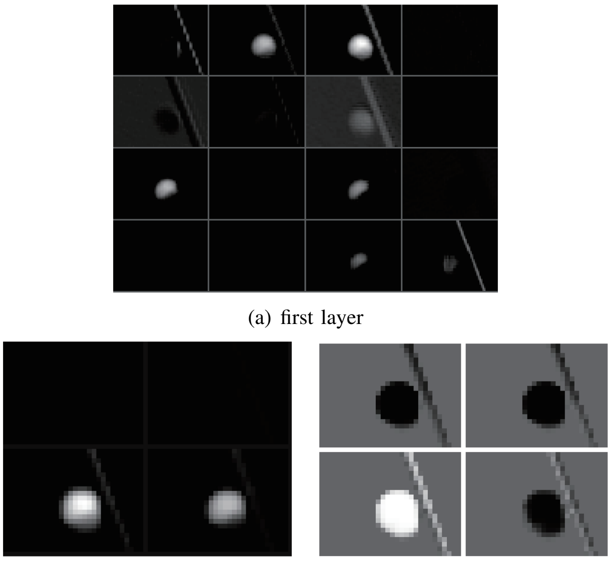
Fig. 7. The feature maps of two convolutional layers pre-trained by classification. White color denotes larger value and black color denotes smaller value accordingly in the output feature maps.

Then the convolutional layers are further fine-tuned by localizing the centroid of ball using a self-learned spatial softmax layer. Fig. 8 shows the feature maps of the two fine-tuned convolutional layers and the corresponding probability distribution maps of softmax. Fig. 8 and Fig. 7 have the same input data. Compared to the feature maps in Fig. 7, the fine-tuned convolutional layers further highlight the ball's feature and suppress the background's distinguishable feature, *i.e.*, the blue table and the white table edge, which obviously contributes to a better localization performance. The ball's two colors contributes equally for feature detection after fine-tuned.

The left-bottom probability distribution map of softmax shows that the pixels that belong to the ball have a larger probability than those belongs to background. However, the other three probability distribution maps are opposite, which the background has nearly the same distribution and the ball has a much smaller probability distribution. Despite the fact that they looks like unreasonable, they help the full convolutional layer effectively decrease the side-effect of background. A more accurate and robust localization result is achieved.

B. Classfication Accuracy

Compared to ImageNet's 1000 classification categories, the classification here is a simple two-category classification



(a) first layer

(b) second layer

(c) softmax layer

Fig. 8. The feature maps of two convolutional layers fine-tuned by localization

task. Using two convolutional layers, it obtains a very high accuracy of classification , as listed in Table I. The result indicates that the convolutional layers are well pre-trained to effectively detect both the ball's and background's features.

TABLE I
CLASSFICATION ACCURACY

	ball	background
ball	99.2%	0.8%
background	0.2%	99.8%

C. Localization

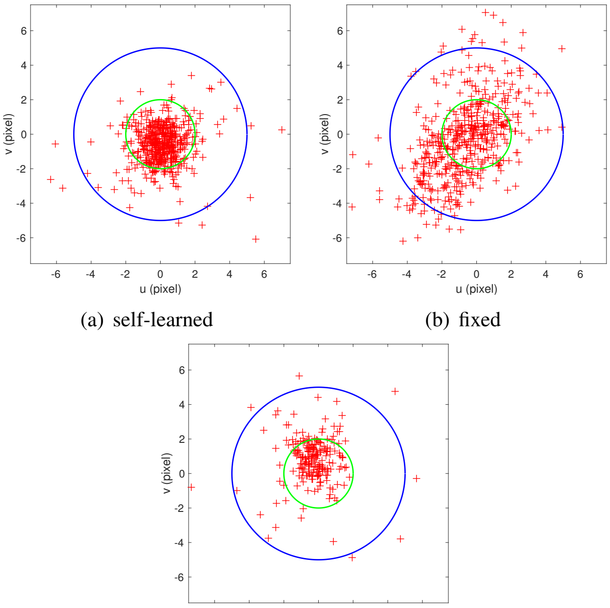
In order to analyze the performance of the proposed learning framework on localization accuracy, we conduct comparison of experiments with the color-based method proposed by Yifeng Zhang [13] as well as the learning framework using a fixed spatial softmax layer [24]. In experiment, we find that the fixed spatial softmax layer is sensitive to the ball's size on the image. Thus the input to the learning framework with a fixed spatial softmax layer is the raw image with resolution of 640×480, because the ball's relative scale on cropped 64×48 image is more variational than on raw image.

500 images are used to test the performance of the three methods. Table II listed the three method's accuracy of detection and localization of the ping-pong. The detection accuracy indicates the chance that the method can successfully detect the ball on image. Since the proposed learning framework always have an output of the ball's centroid no matter whether there is a ball on tested images, we treat it as a failed detection if the localization error is larger than a threshold, 8 pixels. The learning framework with a self-learned or fixed spatial softmax layer achieve much higher detection accuracy than the color-based method, which indicates that the learning framework is more adaptable to different ball's colors and light conditions. Note that the localization accuracy does not count those images that the ball is not successfully detected by the method. As we can see, the learning framework with self-learned spatial softmax layer achieve the best performance in both u and v dimensions.

TABLE II
ACCURACY OF DETECTION AND LOCALIZATION

	detectation	u (pixel)		v (pixel)	
	accuracy	ME	$RMSE$	ME	$RMSE$
Self-Learned	99.2%	0.10	1.34	-0.58	1.33
Fixed	97.4%	-0.68	1.96	-0.35	2.32
Color-Based	44.0%	-0.32	1.40	0.77	1.60

Fit. 9 shows the distribution of the localization error on test data set, which gives us an intuirive illustration of the three methods' performance.



(a) self-learned

(b) fixed

(c) color-based

Fig. 9. The distribution of localization error. The radius of green circle is 2 pixels and the radius of blue circle is 5 pixles. The proposed learning framework with a self-learned spatial softmax layer successfully detects ping-pong ball on 496 images, the learning framework with a fixed spatial softmax layer successfully detects ping-pong ball on 487 images, and the color-based method successfully detects ping-pong ball on 220 images.

We tested the computation efficiency of the proposed learning framework on a platform of a 6G memory and i7 2600K CPU pc running a 64-bit operation system, as shown in Fig. 10. The average computation time of the learning framework is $1.685ms$. If the method runs on a GPU, it could be faster. It definitely satisfies the real-time requirement of robotic table tennis system.

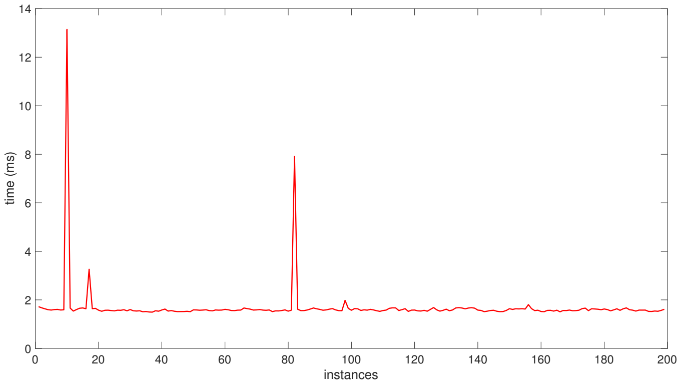


Fig. 10. Computation Time

V. CONCLUSIONS

In this paper, we propose a learning framework that trains a convolutional neural network to detect and localize a ball with high accuracy and robustness. First of all, a convolutional neural network is used to automatically learn the features of ping-pong ball in different colors by classifying the images of background and ping-pong ball. The pre-trained convolutional neural network can effectively detect the ball using learned features. Secondly, we propose a self-learned spatial softmax layer accompanied by a ReLU layer to localize the ball by computing the average centroid of the detected feature pixels. Thirdly, two fully connected layers are further connected to the special softmax layer to regress the balls location on the image by fine-tuning the convolutional layers. The neural network is trained using a large data set of pictures that contains ping-pong ball in different colors, which are collected under various light conditions. The effectiveness and accuracy of the proposed learning framework is verified using the collected data set, in comparison with the state-of-the-art method.

The proposed learning framework is a general framework for object detection and pixel-wise localization. Given a data set, it could learn to detect and localize any objects automatically and effectively. Currently, the learning framework is tested and verified on a single object detection and localization. In the future, the learning framework should be improved to suit the cases of multi-objects detection and localization.

REFERENCES

[1] J. Billingsley, "Robot ping pong," *Practical Computing*, vol. 6, no. 5, 1983.

[2] H. Hashimoto, F. Ozaki, and K. Osuka, "Development of a pingpong robot system using 7 degrees of freedom direct drive arm," in *IECON '87: Industrial Applications of Robotics & Machine Vision*, vol. 0856, 1987, Conference Proceedings, pp. 608–615. [Online]. Available: http://dx.doi.org/10.1117/12.943016

[3] R. L. Andersson, *A robot ping-pong player: experiment in real-time intelligent control*. Cambridge, MA, USA: MIT Press, 1988.

[4] K. Hirota, Y. Arai, and S. Hachisu, "Fuzzy controlled robot arm playing two-dimensional ping-pong game," *Fuzzy Sets and Systems*, vol. 32, no. 2, pp. 149–159, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0165011489902510

[5] F. Miyazaki, M. Takeuchi, M. Matsushima, T. Kusano, and T. Hashimoto, "Realization of the table tennis task based on virtual targets," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4. IEEE, 2002, Conference Proceedings, pp. 3844–3849.

[6] L. Acosta, Rodrigo, J. A. Mendez, Marichal, and M. Sigut, "Ping-pong player prototype," *Robotics & Automation Magazine, IEEE*, vol. 10, no. 4, pp. 44–52, 2003.

[7] K. P. Modi, "Vision application of human robot interaction: development of a ping pong playing robotic arm," Thesis, 2005.

[8] TOSY, "Tosy ping pong playing robot," November 2009. [Online]. Available: https://en.wikipedia.org/wiki/TOPIO

[9] K. Mlling, J. Kober, and J. Peters, "A biomimetic approach to robot table tennis," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. Citeseer, 2010, Conference Proceedings, pp. 1921–1926.

[10] A. Nakashima, Y. Ogawa, C. Liu, and Y. Hayakawa, "Robotic table tennis based on physical models of aerodynamics and rebounds," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. IEEE, 2011, Conference Proceedings, pp. 2348–2354.

[11] Z. Zhang, D. Xu, and M. Tan, "Visual measurement and prediction of ball trajectory for table tennis robot," *Instrumentation and Measurement, IEEE Transactions on*, vol. 59, no. 12, pp. 3195–3205, 2010.

[12] Y. Zhang, W. Wei, D. Yu, and C. Zhong, "A tracking and predicting scheme for ping pong robot," *Journal of Zhejiang University SCIENCE C*, vol. 12, no. 2, pp. 110–115, 2011.

[13] Y. Zhang and R. Xiong, "Real-time vision system for a ping-pong robot," *Scientia Sinica nformationis*, vol. 42, no. 9, pp. 1115–1129, 2012.

[14] L. Hailing, W. Haiyan, L. Lei, K. Kuhnlenz, and O. Ravn, "Ping-pong robotics with high-speed vision system," in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*, 2012, Conference Proceedings, pp. 106–111.

[15] Kuka, "The duell kuka kr agilus vs timo boll," March 2014. [Online]. Available: https://www.youtube.com/watch?v=tIIJME8-au8

[16] J. Glover and L. P. Kaelbling, "Tracking the spin on a ping pong ball with the quaternion bingham filter," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, Conference Proceedings, pp. 4133–4140.

[17] OMRON. (2016, October) About forpheus. [Online]. Available: http://www.omron.com/innovation/forpheus.html

[18] X. Chen, Q. Huang, W. Wan, M. Zhou, Z. Yu, W. Zhang, A. Yasin, H. Bao, and F. Meng, "A robust vision module for humanoid robotic ping-pong game," *International Journal of Advanced Robotic Systems*, vol. 12, 2015.

[19] C. H. Lampert and J. Peters, "Real-time detection of colored objects in multiple camera streams with off-the-shelf hardware components," *Journal of Real-Time Image Processing*, vol. 7, no. 1, pp. 31–41, 2012.

[20] Y. Lecun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, 1989.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, Conference Proceedings, pp. 1097–1105.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Computer Science*, 2015. [Online]. Available: http://arxiv.org/abs/1504.00702v4

[25] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11263-015-0816-y

[26] 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*, 2014, Conference Proceedings, pp. 580–587.

[27] R. Girshick, “Fast r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, Conference Proceedings, pp. 1440–1448.

[28] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Advances in neural information processing systems*, 2015, Conference Proceedings, pp. 91–99.

[29] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*. ACM, 2014, Conference Proceedings, pp. 675–678.