Humanoid Robot Detection using Deep Learning: A Speed-Accuracy Tradeoff

Mohammad Javadi*[1], Sina Mokhtarzadeh Azar*[2], Sajjad Azami*[2], Saeed Shiry Ghidary[2], Soroush Sadeghnejad[1], and Jacky Baltes[3]

[1]Bio-Inspired System Design Lab, Amirkabir University of Technology (Tehran Polytechnic), No. 424, Hafez Ave., Tehran, Iran. P. O. Box 15875-4413.

[2]Cognitive Robotics Lab, Amirkabir University of Technology (Tehran Polytechnic), No. 424, Hafez Ave., Tehran, Iran. P. O. Box 15875-4413.

[3]Department of Electrical Engineering, National Taiwan Normal University, 162 Heping E Road Sec. 1, Taipei, 10610, Taiwan.

correspondingauthor:s.sadeghnejad@aut.ac.ir

Abstract. Recent advances in computer vision have made the detection of landmarks on the soccer field easier for teams. However, the detection of other robots is also a critical capability that has not garnered much attention in the RoboCup community so far. This problem is well represented in different RoboCup Soccer and Rescue Robot Leagues. In this paper, we compare several two-stage detection systems based on various Convolutional Neural Networks (CNN) and highlight their speed-accuracy trade off. The approach performs edge based image segmentation in order to reduce the search space and then a CNN validates the detection in the second stage. We use images of different humanoid robots to train and test three different CNN architectures. A part of these images was gathered by our team and will be publicly available. Our experiments demonstrate the strong adaptability of deeper CNNs. These models, trained on a limited set of robots, are able to successfully distinguish an unseen kind of humanoid robot from non-robot regions.

Keywords: Robot Detection, Robot Vision, Humanoid Robots, Deep Learning, Convolutional Neural Networks, Image Segmentation

1 Introduction

The RoboCup federations ambitious goal for 2050 was stated in 1997: a team of fully autonomous humanoid robot soccer players shall win the soccer game against the winner of the World Cup [1]. In order to reach this purpose, researchers are working on multidisciplinary problems to solve various challenges of creating such intelligent systems. Also, annual RoboCup competitions within

* Authors contributed equally to this work.

different leagues make incremental steps toward this big goal [2]. A crucial part of these systems is extracting information from visual data, i.e., computer vision. A lot of literature has been published on robot vision. Some of them were focusing on mutual identification of robots and more generally, robot body detection. This becomes vital in disaster situations like rescue robots where robots may want to identify each other using camera feed alone, but is also important in soccer where a player must be able to identify team mates and opponents. Humanoid robot teams, in RoboCup Soccer Leagues, must recognize landmarks on the field, e.g., field lines and the goal posts, for localization. But in this paper, we focus on body detection of other robots to present and compare different robust detection systems to detect and classify robot bodies in realistic and complex environments.

Our proposed systems are set to use deep learning methods for validating the results of image segmentation approaches and are expected to stay accurate in different light conditions. Results of this work have been evaluated and tested on three different hardware ranging from a mini computer used by our humanoid robots to a high-speed powerful server equipped with GPU. This leads to a comparison between accuracy and computation speed which is important for robots since there is always a limited computational power available.

The main contributions of this paper are:

1. Using and evaluating three Convolutional Neural Networks with different parameters and iterations to create robust body detection systems for humanoid robots.
2. Using different hardware to provide a speed-accuracy trade off since heterogeneous robots are going to use the system in realistic scenarios.
3. Two step procedure for body detection using image segmentation and CNN.
4. A novel data set of three different humanoid robots captured in realistic conditions and positions from the upper camera of robot in action.
5. Presenting experimental evidence that the proposed system can be used in action by robots in real life scenarios.

The rest of the paper consists of 6 sections: Section 2 presents an overview of methods and approaches in object recognition of robots and Deep Learning classification. Section 3 explains our method in detail. Section 4 contains information about the used dataset, experimental result and evaluation metrics are reported in detail in section 5 and 6. Finally, in section 7, we provide a summary of the work, conclusions and directions for the future work.

2 Related Works

To deal with the vision problem for 2050, the team of robots needs to understand the environment at least as the human team understands it. Limited dynamic range of cameras, changes in colour due to brightness, and distortions due to motion make it impossible to create a robust system which classifies robot bodies using raw color information only. In this section, we briefly review the previous

works on image segmentation and robot detection. For image segmentation, despite the fact that color segmentation is common in RoboCup Soccer Leagues [3], Ma et al. [4] presented an approach as the edge flow which facilitates the integration of color and texture for this purpose. Fan et al. [5] integrated the results of color-edge extraction and SRG (seeded region growing) to provide homogeneous image regions with accurate and closed boundaries. On the other hand, Ren et al. [6] presented Region Proposal Networks (RPNs) that simultaneously predicts the object bounds and objectness scores at each position. In the case of robot detection and object recognition, Sabe et al. [7] focused on obstacle detection by plane extraction from data captured by stereo vision. In another work, Farazi et al. [8] used color segmentation of the black color range for obstacle detection. They also implemented a robot detector using a HOG feature descriptor in the form of a cascade classifier for detection of the igus® Humanoid Open Platform [9]. Arenas et al. [10] used a nested cascade of boosted classifiers for detection of Aibo robot and humanoid robots. Shangari et al. [11] evaluated combinations of cascade classifier with Histograms of Oriented Gradients, Local Binary Patterns, and Haar-like features for Humanoid robots detection and believed LBP feature is more useful than the others. Albani et al. [12] used a Deep Learning approach for NAO robot detection in the RoboCup Standard Platform League (SPL). Here, we show that this approach can be extended to deal with different types of robots and be used in other RoboCup leagues.

3 Proposed Approach

Fig. 1 shows an overview of our system. A wide-angle YUV422 image is the input of our system. A human first trains the system by selecting seed pixels to create a look up table for color classification. This table is a mapping from YUV color space to a set of specific colors and assigns a class label to each pixel.

To use edge based image segmentation and Hough Transform algorithms, we compute a binary image which describes edge intensity of each pixel in a given raw image. A grayscale image is generated by extraction of the Y channel. Afterwards, we compute the Scharr gradient operator(explained in [20]) on the grayscale image which results in the desired binary image. The new images compared to the camera image can be seen in Fig. 2.

3.1 Segmentation and Bounding Box extraction

Reduction of the search space can increase the performance of the whole system, both in terms of time and accuracy. In order to find regions of interestes (ROI), firstly a vertical scan line runs inside the binary image in order to find pixels with high edge intensity. These edge spots builds ROI boundaries and have the potential to construct a same shape. Then, based on two different feature types we find related spots and connect them to build boundary of objects like ball, lines, goals and obstacles. The first feature is euclidean distance of selected spots in the X and Y direction, And the second one is the size and color of the area

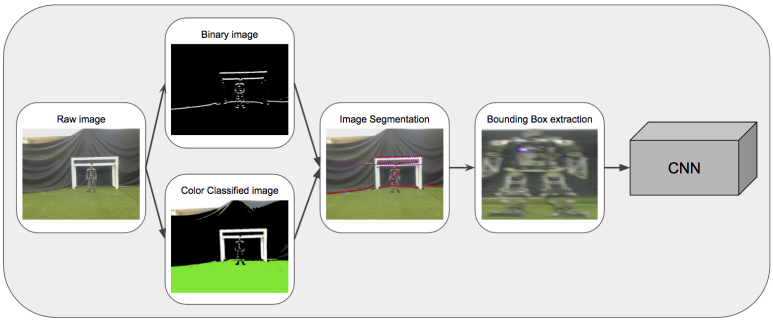

Fig. 1. Overview of our system.

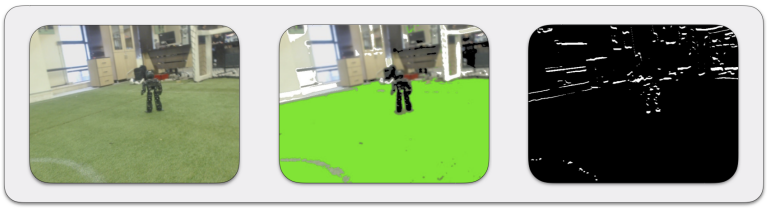

Fig. 2. Left: system input in RGB format. Middle: color classified image. Right: binary image.

around each spot that can help identify which object the spot belongs to. For example, in the case of obstacle detection, after perception of the black area inside the big green space and due to the fact that robots should have black feet in the RoboCup humanoid league, it can be concluded that considered boundary belongs to robot feet. Our algorithm moves from the region of detected feet to left and right until a continues green region according to a threshold is seen. To extract the proposed region for robot, we crop a rectangle from foot to horizontal line(computed from robot structure) in Y axis and from left most point to right most point in X axis. Regions inside other bigger bounding boxes are omitted from set of proposed regions(see Fig. 3).

3.2 Validation

As shown in Fig. 3, regions extracted in previous section may contain false positives. Similar to work in [12] we have a validation step in which a CNN is used to omit irrelevant outputs from detected regions. Three different architectures were used in our work, namely LeNet, SqueezeNet and GoogLeNet. These models were chosen because of their computational efficiency. The training of deep convolutional neural networks requires a lot of data and computational power. Preparing this amount of training data for every specific task is very time consuming and even may not be possible in some domains. Furthermore, not every one has access to high end GPUs to train these models. To solve this problem,

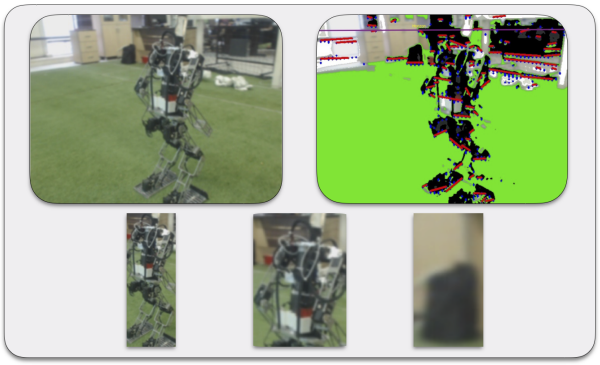

Fig. 3. Top Left: input image. Top Right: color classified image. Edge spots are blue colored, related spots are connected with red lines. Bottom Left: true positive robot region. Bottom Middle: proposed region is inside the bottom left region and will be ignored. Bottom right: false positive robot region.

pretrained networks on other datasets with different tasks are fine-tuned on a new relatively small dataset for a new task. Here we fine-tune SqueezeNet and GoogLeNet architectures that are pretrained on the Imagenet Dataset, for our validation task.

3.3 CNN Architectures

A brief overview of all three architectures is in Fig. 4. First architecture we used is a variant of the LeNet architecture[18]. Main difference is the number of output filters in first convolution, second convolution and first fully connected layer which are set to 20, 50 and 500 respectively. Also one fully connected layer is deleted. GoogLeNet is the second architecture used in this work. This model is a deep Convolutional Neural Network which makes use of layers called Inception modules[14]. Most recent architecture that focuses on decreasing model parameters is the work of Iandola et al. [17]. In a similar approach to [14] fire modules are used to construct the model. In their work same level accuracy as AlexNet[22] is achieved with 50x fewer parameters. This model is both fast and accurate which makes it a good choice to run on humanoid robots.

4 Dataset Description

To train our models we gathered images from three robots(see Fig. 5 for samples). 500 images of each robot were selected. Additionaly, 500 images from SPQR NAO Image dataset[12] were chosen randomly. We also used 2000 random nonrobot images from their dataset. These 2000 robot and 2000 nonrobot images were used as our dataset. We make images of our humanoid robots publicly available and encourage others to add images of their robots to help gather a large dataset of different kinds of humanoids for future research.

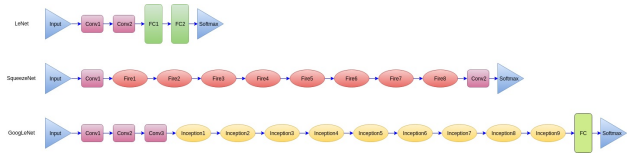

Fig. 4. View of three architectures with convolution and fully connected layers visible only. Depth of an architecture is the count of convolution and FC layers. Fire and Inception modules has depth of 2 convolutions. Therefore LeNet, SqueezeNet and GoogLeNet have depths of 4, 18 and 22 respectively.

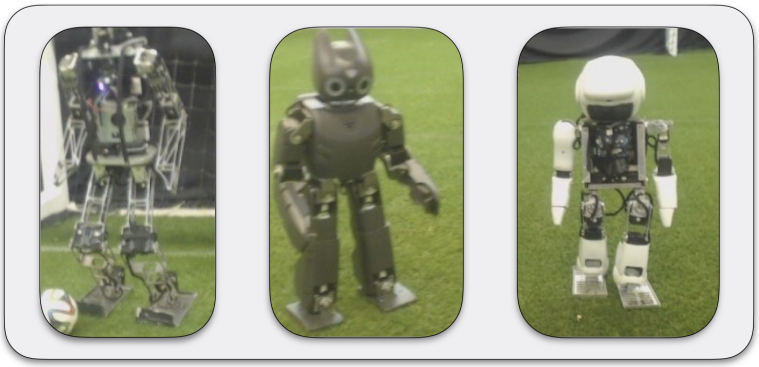

Fig. 5. Sample images of the used dataset. The left one is a sample image of ARASH. In middle we have DARWIN and the right one is KIARASH.

5 Implementation Details for CNNs

Our implementation is based on Caffe[15]. Caffe is a deep learning framework. Training parameters are mainly the same as in original papers. Details are in Table 1. We trained LeNet from scratch. GoogLeNet and SqueezeNet were fine-tuned. SqueezeNet v1.1 was chosen from different versions of SqueezeNet because of less computation. We used a Nvidia Geforce GTX 980 with 4 gigabytes of RAM to train our models.

Table 1. Training parameters details. Batch size for SqueezeNet and GoogLeNet were limited due to lack of memory. Learning Rate is multiplied by gamma every stepsize.

	LeNet	SqueezeNet	GoogLeNet
Iteration	8000	2000	2000
Batch Size	256	64	32
Base Learning Rate	0.001	0.001	0.0001
Learning Rate Policy	step	step	step
Step Size	1000	500	500
Gamma	0.1	0.1	0.1
Momentum	0.9	0.9	0.9

6 Experimental Results

In our validation step, we aim to distinguish between robot and non-robot regions. Therefore the problem is modeled as a binary classification problem. Different experiments were conducted to evaluate performance of all three models. First, robot and non-robot images were shuffled separately and 66% of each were chosen for training, the rest were chosen for the test set. Due to randomness in training procedure, each model was trained 5 times and average accuracy of those models on the test set is reported. According to results in Table 2, all models perform well in this dataset(LeNet a bit lower than the other two). Also, the variance of accuracies in all architectures is reasonably low.

Time needed for every forward pass of all three architectures is reported in Table 3. Our measurements were on two different GPUs(GTX 980 with 2048 cuda cores and GT 620m with 96 cuda cores), a general purpose laptop CPU and a humanoid robot CPU. As expected, LeNet is faster than deeper architectures by a large margin. SqueezeNet is faster than GoogLeNet and runs in a reasonable time on Humanoid robot's system. Time for a forward pass of GoogLeNet increases dramatically when using a CPU only.

Table 2. Average and maximum accuracy of 5 models of each architecture on the test set. Images of all robots in train and test set.

	Average Accuracy	Max Accuracy
LeNet	94.83	97.94
GoogLeNet	99.90	100
SqueezeNet	100	100

Table 3. Average test time of every architecture on different platforms.

	GTX980(ms)	GT620m(ms)	Core i5 2.50GHz(ms)	Celeron Dual Core 1.10GHz(ms)
LeNet	0.25	0.76	2.36	4.17
GoogLeNet	6.97	48.77	602.73	1754.03
SqueezeNet	2.13	16.63	48.55	173.33

We set two other experiments to compare the efficiency of the three architectures in a more difficult situations. First, all Images of ARASH were seperated from other robots and were given as test set. 500 random non-robot images were also added to test set. Trained models on images from other three robots showed promising results on test set(see Table 4). Same experiment was conducted us-

Table 4. Average and maximum accuracy of 5 models of each architecture on test set. No images of ARASH in training set.

	Average Accuracy	Max Accuracy
LeNet	97.56	98.9
GoogLeNet	99.4	99.8
SqueezeNet	99.3	99.3

ing the Robotis OP robot. This time average accuracy of LeNet decreased about 20%(see Table 5). Other two deeper models were still performing well on test set. We can see that SqueezeNet and GoogLeNet can be trained on Images of a limited types of robots and distinguish between unseen new robot and non-robot regions. This is closer to capabilities of human vision(we can distinguish between robot and non-robot by seeing samples from two or three kinds of robots). Due

to high variance of LeNet on this experiment, we would like to report a 95% normal-based Confidence Interval of accuracy, which is equal to (56.4, 96.4).

Table 5. Average and maximum accuracy of 5 models of each architecture on test set. No images of Darwin in training set.

	Average Accuracy	Max Accuracy
LeNet	76.4	97.3
GoogLeNet	99.4	100
SqueezeNet	100	100

To further evaluate discriminative power of models we changed the problem to a multiclass classification problem. Five classes of non-robot, Arash, Robotis-OP, kiarash and NAO were considered. 500 images from every class were chosen. We used 60% of data for training and other 40% for testing the models. Base Learning rate for GoogLeNet was set to 0.00001. Average accuracy of models in table 6 shows that increasing difficulty of the problem by increasing number of classes lowers the accuracy of weaker models. GoogLeNet's discriminative power can be higher than SqueezeNet if proper amount of data is available. Here, due to low amount of data and higher number of parameters in GoogLeNet relative to SqueezeNet the former shows poor performance relative to latter. Overally, due to superior performance of SqueezeNet in speed and accuracy, this model is a good choice for difficult classification tasks in a humanoid robot.

Table 6. Average and maximum accuracy of 5 models of each architecture on test set in multiclass problem.

	Average Accuracy	Max Accuracy
LeNet	54.06	58.80
GoogLeNet	86.46	89.00
SqueezeNet	98.60	98.60

For more precise comparison, confusion matrices of multiclass experiment is reported below in fromula 1, 2, and 3 noted as CM. Also, precision, recall, and f1-score are reported in tables 7, 8, 9 respectively.

$$CM_{GoogLeNet} = \begin{array}{c} \\ nonrobot \\ ARASH \\ DARWIN \\ KIARASH \\ NAO \end{array} \begin{array}{c} \begin{array}{ccccc} nonrobot & ARASH & DARWIN & KIARASH & NAO \end{array} \\ \begin{bmatrix} 186 & 4 & 2 & 4 & 4 \\ 0 & 132 & 0 & 16 & 52 \\ 3 & 4 & 188 & 4 & 1 \\ 0 & 5 & 20 & 174 & 1 \\ 1 & 0 & 2 & 11 & 186 \end{bmatrix} \end{array} \quad (1)$$

$$CM_{SqueezeNet} = \begin{array}{c} \\ nonrobot \\ ARASH \\ DARWIN \\ KIARASH \\ NAO \end{array} \begin{array}{c} \begin{array}{ccccc} nonrobot & ARASH & DARWIN & KIARASH & NAO \end{array} \\ \begin{bmatrix} 200 & 0 & 0 & 0 & 0 \\ 0 & 186 & 0 & 1 & 13 \\ 0 & 0 & 200 & 0 & 0 \\ 0 & 0 & 0 & 200 & 0 \\ 0 & 0 & 0 & 0 & 200 \end{bmatrix} \end{array} \quad (2)$$

$$CM_{LeNet} = \begin{array}{c} \\ nonrobot \\ ARASH \\ DARWIN \\ KIARASH \\ NAO \end{array} \begin{array}{c} \begin{array}{ccccc} nonrobot & ARASH & DARWIN & KIARASH & NAO \end{array} \\ \begin{bmatrix} 49 & 3 & 144 & 4 & 0 \\ 0 & 129 & 4 & 67 & 0 \\ 0 & 0 & 195 & 5 & 0 \\ 0 & 5 & 2 & 193 & 0 \\ 1 & 7 & 128 & 64 & 0 \end{bmatrix} \end{array} \quad (3)$$

Table 7. Evaluation results for GoogLeNet.

Class	Precision	Recall	F1-Score	Support
nonrobot	0.98	0.93	0.95	200
ARASH	0.91	0.66	0.77	200
DARWIN	0.89	0.94	0.91	200
KIARASH	0.83	0.87	0.85	200
NAO	0.76	0.93	0.84	200
Avg/Total	0.87	0.87	0.86	1000

7 Conclusion

In this study, we presented and tested various real time two-stage vision systems for humanoid robot body detection with promising results. Our pipeline

Table 8. Evaluation results for SqueezeNet.

Class	Precision	Recall	F1-Score	Support
nonrobot	1.00	1.00	1.00	200
ARASH	1.00	0.93	0.96	200
DARWIN	1.00	1.00	1.00	200
KIARASH	1.00	1.00	1.00	200
NAO	0.94	1.00	0.97	200
Avg/Total	0.99	0.99	0.99	1000

Table 9. Evaluation results for LeNet.

Class	Precision	Recall	F1-Score	Support
nonrobot	0.98	0.24	0.39	200
ARASH	0.90	0.65	0.75	200
DARWIN	0.41	0.97	0.58	200
KIARASH	0.58	0.96	0.72	200
NAO	0.00	0.00	0.00	200
Avg/Total	0.57	0.56	0.48	1000

performs a preprocessing stage to reduce search space and a CNN validates the results of ROI detection. The variety of structures and colors in humanoid robots(unlike standard platforms)demands more flexible detection systems than using color segmentation alone, so it makes sense to use edge based image segmentation. Also, this approach performs well in different lighting conditions in the context of time and accuracy. For validation step, we used three different CNNs and tested each on three different hardware, comparison of these systems lets us choose the proper system regarding our application and available computational power. Also, a novel dataset of humanoid body images is published with this project. All the images and sample codes are available on project's repository under https://github.com/AUTManLab/HumanoidBodyDetection. As future works, we are planning to develop a fast and complete humanoid body classifier using a more customized CNN architecture.

8 Acknowledgements

This research is supported by a grant to Jacky Baltes from the Center of Learning Technology for Chinese and the Aim for the Top University Project of the National Taiwan Normal University (NTNU) in Taipei, Taiwan. The research is also supported through a grant to Jacky Baltes by the Ministry of Education, Taiwan, and Ministry of Science and Technology, Taiwan, under Grants no. MOST 105-2218-E-003 -001 -MY2.

References

1. Gerndt, R., Seifert, D., Baltes, J.H., Sadeghnejad, S. and Behnke, S., 2015. Humanoid robots in soccer: Robots versus humans in RoboCup 2050. IEEE Robotics & Automation Magazine, 22(3), pp.147-154.
2. Baltes J, Sadeghnejad S, Seifert D, Behnke S. RoboCup humanoid league rule developments 20022014 and future perspectives. InRobot Soccer World Cup 2014 Jul 15 (pp. 649-660). Springer International Publishing.
3. Rőfer T, Laue T, Műller J, Bartsch M, Batram MJ, Bőckmann A, Bőschen M, Kroker M, Maaß F, Műnder T, Steinbeck M. B-Human team report and code release 2013 (2013), only available online: http://www. b-human. de/downloads/publications/2013. CodeRelease2013. pdf.
4. Ma WY, Manjunath BS. EdgeFlow: a technique for boundary detection and image segmentation. IEEE transactions on image processing. 2000 Aug;9(8):1375-88.
5. Fan J, Yau DK, Elmagarmid AK, Aref WG. Automatic image segmentation by integrating color-edge extraction and seeded region growing. IEEE transactions on image processing. 2001 Oct;10(10):1454-66.
6. Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. InAdvances in neural information processing systems 2015 (pp. 91-99).
7. Sabe K, Fukuchi M, Gutmann JS, Ohashi T, Kawamoto K, Yoshigahara T. Obstacle avoidance and path planning for humanoid robots using stereo vision. In obotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on 2004 Apr (Vol. 1, pp. 592-597). IEEE.

8. Farazi H, Allgeuer P, Behnke S. A monocular vision system for playing soccer in low color information environments. In roceedings of 10th Workshop on Humanoid Soccer Robots, IEEE-RAS Int. Conference on Humanoid Robots, Seoul, Korea 2015.
9. Farazi H, Behnke S. Real-Time Visual Tracking and Identification for a Team of Homogeneous Humanoid Robots, In Proceedings of 20th RoboCup International Symposium, Leipzig, Germany, July 2016.
10. Arenas M, Ruiz-del-Solar J, Verschae R. Detection of aibo and humanoid robots using cascades of boosted classifiers. InRobot Soccer World Cup 2007 May 25 (pp. 449-456). Springer Berlin Heidelberg.
11. Shangari TA, Shams V, Azari B, Shamshirdar F, Baltes J, Sadeghnejad S. Inter-humanoid robot interaction with emphasis on detection: a comparison study. The Knowledge Engineering Review. 2017 Feb;32.
12. Albani D, Youssef A, Suriani V, Nardi D, Bloisi DD. A Deep Learning Approach for Object Recognition with NAO Soccer Robots, July 2016.
13. Shangari TA, Shamshirdar F, Heydari MH, Sadeghnejad S, Baltes J, Bahrami M. AUT-UofM humanoid TeenSize joint team; A new step toward 2050s humanoid league long term RoadMap. InRobot Intelligence Technology and Applications 3 2015 (pp. 483-494). Springer International Publishing.
14. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2015 (pp. 1-9).
15. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T. Caffe: Convolutional architecture for fast feature embedding. InProceedings of the 22nd ACM international conference on Multimedia 2014 Nov 3 (pp. 675-678). ACM.
16. Felzenszwalb PF, Huttenlocher DP. Efficient graph-based image segmentation. International journal of computer vision. 2004 Sep 1;59(2):167-81.
17. Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size. arXiv preprint arXiv:1602.07360. 2016 Feb 24.
18. LeCun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 1998 Nov;86(11):2278-324.
19. Jiang X, Bunke H. Edge detection in range images based on scan line approximation. Computer vision and image understanding. 1999 Feb 1;73(2):183-99.
20. Levkine G. Prewitt, Sobel and Scharr gradient 5x5 convolution matrices. Image Process. Articles, Second Draft. 2012.
21. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. InComputer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on 2009 Jun 20 (pp. 248-255). IEEE.
22. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. InAdvances in neural information processing systems 2012 (pp. 1097-1105).