RoboCup 2020 Humanoid League Team Descriptions, Bordeaux, France, June 2020.

# NimbRo AdultSize Software Description 2020

Hafez Farazi, Grzegorz Ficht, Diego Rodriguez, Dmytro Pavlichenko, Mojtaba Hosseini, Oleg Kosenko, Marcell Missura and Sven Behnke

Rheinische Friedrich-Wilhelms-Universität Bonn
Computer Science Institute VI, Autonomous Intelligent Systems
Endenicher Alle 19A, 53115 Bonn, Germany

{ farazi | ficht | rodriguez | behnke } @ ais.uni-bonn.de
http://www.nimbro.net

**Abstract.** This paper describes the state of robots' software of team NimbRo AdultSize of the Rheinische Friedrich-Wilhelms-Universität Bonn, Germany. This paper aims to serve as qualification material for the competition held in Bordeaux from June 23–29. The design and construction of our robots were developed entirely by our team members. This paper focuses mainly on the software components that played a key role in winning the Humanoid League 2019 in Sydney. These components include: behavior design, robot perception, and gait optimization.

## 1 Introduction

Each year, the competition rules are adapted to resemble more real soccer scenarios and to be more compliant with FIFA rules. This enforces the teams participating in the Humanoid League to improve drastically both the hardware and software capabilities of their platforms. In 2019, for example, the number of

**Fig. 1.** The NimbRo team at RoboCup 2019 in Sydney, Australia.

robot players in the AdultSize category has increased together with the dimensions of the field, which encourages the team to build new robot platforms and to develop teamplay strategies.

In the Humanoid League, team NimbRo has a long history. From 2009 until 2013, our TeenSize robots won the tournament each year consecutively, also obtaining the technical challenges trophy in the years 2012 and 2014. In 2015, we won the RoboCup Design Award. The following year, our team was awarded the first International HARTING Open Source Prize [3], and won TeenSize league [8]. In 2017, team Nimbro held the title in its last participation in the TeenSize category [15]. In the same year, the AdultSize category introduced regular one vs. one games [11], which encouraged team NimbRo to participate in this category, which resulted in winning the AdultSize tournament, drop-in games and technical challenges. Additionally, the newly designed NimbRo-OP2 won the RoboCup Design Award. In 2018, team Nimbro AdultSize received all of the possible awards by winning the main tournament, drop-in games, technical challenges and finally the Best Humanoid Award (Fig. 1) In RoboCup 2019 in Sydney, team NimbRo repeated this feat, and held on to all four possible titles.

For the 2020 competition, we want to exhibit more advanced team play strategies and demonstrate our improvements made in our open-source ROS framework, especially in the areas of perception, localization, walking, and soccer behaviors.

## 2 Perception

### 2.1 Computer Vision

Our visual perception pipeline improved significantly since RoboCup 2018. Thanks to our new unified perception convolutional neural network (NimbRoNet2), we now can reliably perceive the environment in extremely low and very bright lighting condition. The visual perception system can recognize soccer-related objects, including a soccer ball, field boundaries, robots, line segments, and goalposts through the usage of texture, shape, brightness, and color information.

Our deep-learning-based visual perception system is robust against brightness, viewing angles, and lens distortions. To achieve this, we designed a unified deep convolutional neural network to perform object detection and pixel-wise classification with one forward pass. After post-processing, we managed to outperform our previous non-deep learning approach to soccer vision [7] as well as our previous deep-learning-based model [12]. Our perception system is also able to track [9] and identify our robots [10].

The system has two output heads; one for object detection, and the other for pixel-wise segmentation. The detection head gives the location of the ball, robots, and goalposts. The segmentation head is for line and field detection. Our model uses an encoder-decoder architecture similar to pixel-wise segmentation models like SegNet [4], and U-Net [17]. Due to computational limitations and the necessity of real-time perception, we have made several adaptations, e.g., using a shorter decoder than the encoder. Thus, the number of parameters has been reduced for the cost of losing fine-grained spatial information which can be alleviated using sub-pixel post-processing. To minimize annotation efforts, we utilized transfer-learning. A pre-trained ResNet-18 is chosen as the encoder. Since ResNet was originally designed for recognition tasks, we removed the Global Average Pooling (GAP) and the fully connected layers in the model. Transpose-convolutional layers are used for up-sampling the representations. To use location-dependent features, we used newly proposed location-dependent convolutional layer [14]. In order to limit the number of parameters used, a shared learnable bias between both output heads is implemented. The proposed visual perception architecture is illustrated in Fig. 2.

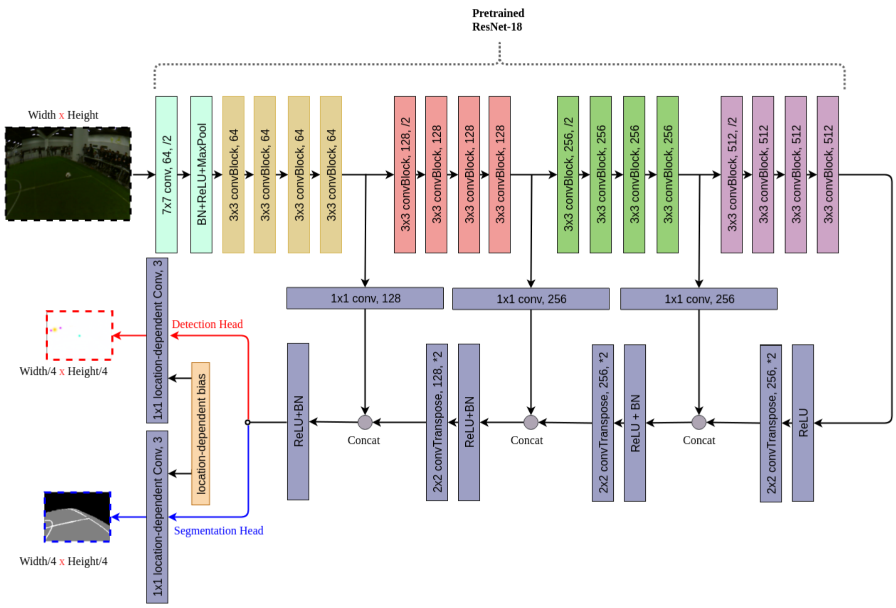


**Fig. 2.** NimbRoNet2 architecture. Similar to ResNet, each convBlock consists of two convolutional layers followed by batch-norm and ReLU activations. For simplicity, residual connections in ResNet are not depicted. Note that instead of a convolutional layer we used a location-dependent convolution in the last layer.

Different losses were used for different network heads. For detection head, similar to SweatyNet [18], the mean squared error is employed. The target is constructed by Gaussian blobs around the ball center and bottom-middle points of the goalposts and robots. In contrast to last year model, NimbRoNet2 uses a bigger radius for robots with the intuition that annotating a canonical center point is more difficult, thus a bigger radius would less penalize the network for not outputting the exact human labels. In the classification head, we used pixel-wise Negative Log Likelihood. We also added Total Variation loss to the output of all result channels except the line segmentation channel. Total Variation loss encouraged blob response thus helped to have less false positives, especially in field detection.

One other difficulty of this year of RoboCup was very thin goalposts which were hard to detect. However, with many training samples, the network finally managed to learn it very robustly. After sufficient training, goal posts were detected even when they were hard to recognize by a human. This might be explained by inferring their presence from other features of the pitch like field boundary and lines. One detected hard-to-recognize goal post is shown in the last row of Fig. 3.

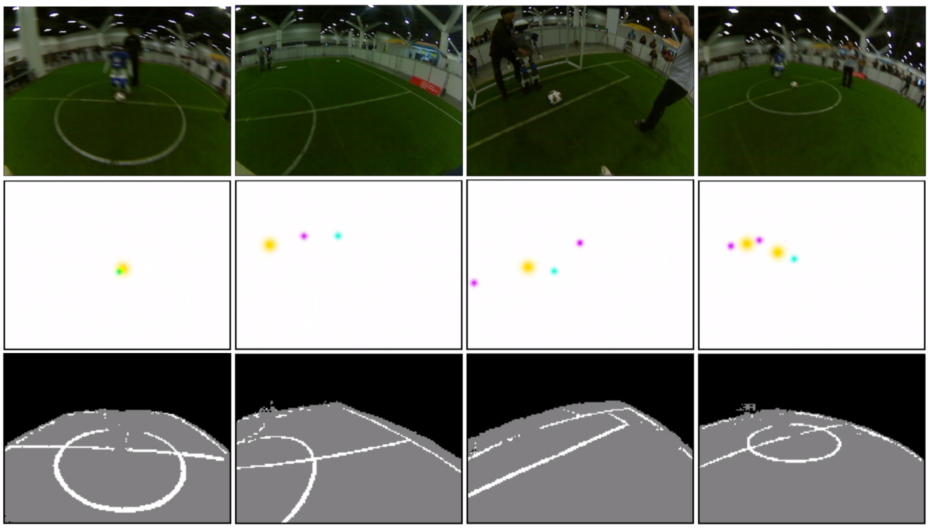

**Fig. 3.** Object detection results. Upper row: captured images by our robots. Middle row: the output of the network with balls (cyan), goal posts (magenta), and robots (yellow). Bottom row: the output of the segmentation branch with lines (white), field (gray), and background (black).

Despite using Adam optimizer, which has an adaptable per-parameter learning rate, finding a suitable learning rate is still a challenging prerequisite for training. To determine an optimal learning rate, we followed the approach presented by Smith et. al. [19]. Each batch contained only some samples for one of the output heads. We used progressive image resizing that uses small pictures at the beginning of training, and step by step increase the dimensions as training progresses, a method inspired by Brock et. al. [6] and by Yosinski et. al. [20]. In early iterations, the inaccurate randomly initialized model will make fast progress by learning from large batches of small pictures. Within the initial fifty epochs, we used downsampled training images, whereas the weights on the encoder part are frozen. Throughout the following fifty epochs, all parts of the models are jointly trained. In the last fifty epochs, full-sized pictures are used to learn fine-grained details. A lower learning rate is employed for the encoder

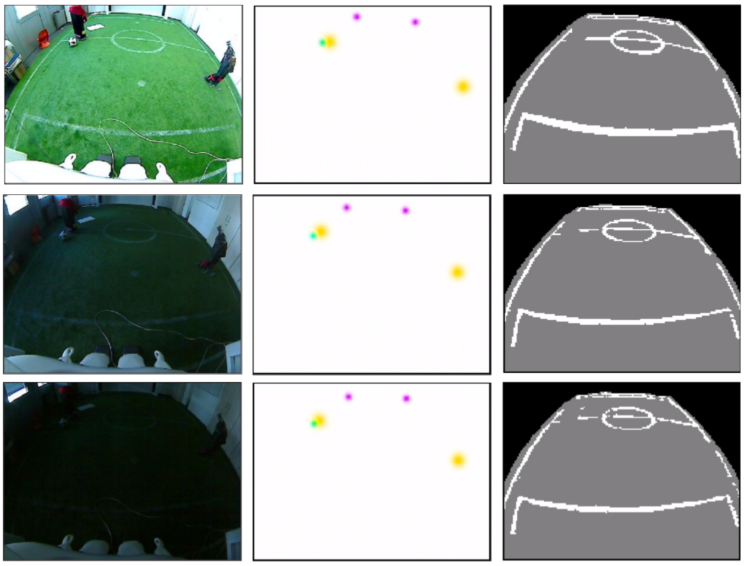

**Fig. 4.** Object detection under various lighting conditions. Left column: captured images by our robots. Middle column: output of the network indicating balls (cyan), goal posts (magenta), and robots (yellow). Right column: output of the segmentation branch showing lines (white), field (gray), and background (black).

part, with the intuition that the pre-trained model needs less training time to converge. With the described method, the entire training process with around 9k samples takes less than three hours on a single Titan X GPU with 12 GB of memory. Examples from the test set are pictured in Fig. 3. To annotate more data as quickly as possible, we designed an annotation tool which automatically annotates the input based on the previously trained model. The user then only had to correct those samples which were wrongly classified. This semi-automatic annotation tool was crucial for us to gather as many samples as possible from the RoboCup 2019 environment.

The output of the network is of lower resolution and has less spatial information than the input image. To account for this effect in the detection part, we calculate sub-pixel level coordinates based on the center of mass of a detected contour. There was no need to account for lower resolution output in the field and line segmentation.

After detecting soccer-related objects, we filter them and project each object location into egocentric world coordinates. Using NimbRoNet2, we can detect objects which are up to 10 meters away. The complete perception pipeline, including a forward-pass of the network, takes approximately 36 ms on the robot hardware. Using a unified network helped both detection and segmentation. The network learned to exclude the balls which were outside of the field, hence reducing false detection rate. Outside field object removal was previously done only after post-processing. In addition, the robot was able to play soccer in pitch black, and the perception was robust in different lighting conditions, including

**Table 1.** Results of the detection branch of our visual perception network.

| Type | F1 | Accuracy | Recall | Precision | FDR |
|---|---|---|---|---|---|
| Ball (NimbRoNet2) | **0.998** | **0.996** | 0.996 | **1.0** | **0.0** |
| Ball (NimbRoNet) | 0.997 | 0.994 | **1.0** | 0.994 | 0.005 |
| Ball (SweatyNet-1 [18]) | 0.985 | 0.973 | 0.988 | 0.983 | 0.016 |
| Goal (NimbRoNet2) | **0.981** | **0.971** | 0.973 | **0.988** | **0.011** |
| Goal (NimbRoNet) | 0.977 | 0.967 | **0.988** | 0.966 | 0.033 |
| Goal (SweatyNet-1 [18]) | 0.963 | 0.946 | 0.966 | 0.960 | 0.039 |
| Robot (NimbRoNet2) | **0.979** | **0.973** | **0.963** | **0.995** | **0.004** |
| Robot (NimbRoNet) | 0.974 | 0.971 | 0.957 | 0.992 | 0.007 |
| Robot (SweatyNet-1 [18]) | 0.940 | 0.932 | 0.957 | 0.924 | 0.075 |
| Total (NimbRoNet2) | **0.986** | **0.986** | 0.977 | **0.994** | **0.005** |
| Total (NimbRoNet) | 0.983 | 0.977 | **0.982** | 0.984 | 0.015 |
| Total (SweatyNet-1 [18]) | 0.963 | 0.950 | 0.970 | 0.956 | 0.043 |

**Table 2.** Results of the semantic segmentation of our visual perception network.

| Type | Accuracy | IOU |
|---|---|---|
| Field | 0.986 | 0.975 |
| Lines | 0.881 | 0.784 |
| Background | 0.993 | 0.981 |
| Total | 0.953 | 0.913 |

direct sunlight and without ambient light (Fig. 4). Unfortunately, this year, all AdultSize games were played with artificial light, thus we could not test our new development for lighting conditions during the competition.

Our visual perception pipeline is compared on different soccer-related objects against SweatyNet [18] and our previous model NimbRoNet [12] (Table. 1). We also evaluated our segmentation head (Table. 2). We have outperformed SweatyNet and NimbRoNet, whose results were one of the best-reported in terms of detecting soccer objects. This achievement was also accompanied by being approximately two times faster than SweatyNet in training phase. The reduced training time can be attributed to the progressive image resizing and transfer learning techniques.

**Localization:** Localization of the robot on the soccer field—the task of estimating the 2D pose $(x, y, \theta)$ of the robot—is performed using the field line, center circle and goal post detection. Each component of the 2D pose is estimated independently. To estimate the $\theta$ component, we keep track of initial orientation and maintain an internal correction term based on the angular deviation between the expected and detected orientations of the white lines. This approach does not rely on having an accurate gyroscope output, and in experiments was able to correct deviations up to 10° coming from the gyroscope. Using the estimated $\theta$, which is normally quite exact, we can rotate every vision detection to align with the global field coordinate system. The detected line segments can thereby be classified as being either horizontal or vertical field lines. In each cycle of the localization node, we use the perception information and dead-reckoning walking data to update the previously estimated 2D location. For updating 2D location, we distinguish $x$ and $y$ component using estimated $\theta$. The $y$ component of the localization is updated based on the $y$-components of the detected center circle, goal posts, and vertical field lines. With the assumption that the robot is always inside the field lines, the vertical sidelines can easily be differentiated and used for updates. The $x$-component of the localization is analogously updated based on the $x$-components of the detected center circle, goal posts and horizontal field lines. The horizontal lines belonging to the goal area are discriminated from the center line by checking for the presence of a consistent goal post detection, center circle detection, and/or further horizontal line that is close and parallel. This approach can easily deal with common localization difficulties, such as sensor aliasing and robot kidnapping. In contrast to some other proposed localization methods for soccer fields, this method is relatively easy to implement and very robust.

To solve the global localization problem, our method relies on having a source of global yaw rotation of the robot. In our robots, gyro integration is a reliable source of orientation tracking, but it needs a global reference. In order to set the initial heading, we could either use manual initialization or automatic initial orientation estimation. Although manual initialization can be done once before the start of each game, it can fail during the match. Sometimes restarting the operating system of the robot is unavoidable, which will force a reinitialization of the heading. As a result, we reformulated the heading initialization problem as a classification task.

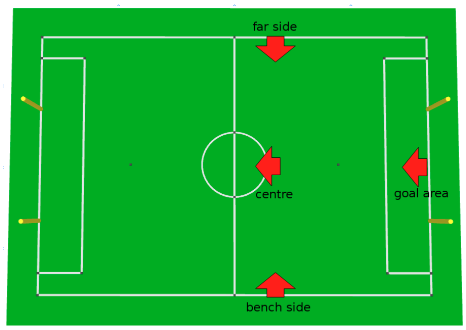


**Fig. 5.** Set of predefined positions the robot can start in.

According to the rules, there are a few predefined positions and orientations that the robot can start in or enter the game from. As shown in Fig. 5, the robot can start in four different positions. In two of the spots, it should face the opponent's goal—near the center circle and goal area. The other two sets of locations are at the sideline in the robot's own half—facing the field. We employ a multi-hypothesis version of our localization module, which is initialized with four instances of initial hypothetical locations. During a brief period at the beginning, the robot tries to find the most probable hypothesis among all running instances. This stops when either the process times out or the robot finds the best hypothesis. Finally, the vision module keeps the best instance and discards the rest. To make sure that the decision is correct, we double-check the result based on the perceived landmarks like goalposts and the center circle.

**Gait Optimization:** The gait of our robots is based on an open-loop pattern generator that calculates joint states based on a gait phase angle, whose rate is proportional to the desired step frequency [5]. The phase angle is responsible for generating arm and leg movements such as lifting and swinging. We have built around this approach and incorporated corrective actions based on fused angle feedback [1, 2, 13].

We use Bayesian optimization methods to find proper values for the Fused Feedback controller used by the gait. To minimize hardware wear-off, this optimization does not only take place in the real world, but highly exploits information gained through the included Gazebo simulator. This approach has been previously applied to the igus® Humanoid Open Platform robot and is now utilized on the AdultSize platforms [16].

## 3 Behavior and Teamplay

Teams participating in the AdultSize class in RoboCup 2019 were composed of a maximum of two robots. We define dynamic *Player Tasks*, which are frequently reassigned during the game. This task tells the robot what it is supposed to do according to its own state in the field and the state of its teammates. In addition, we define a task manager which is in charge of the safe assignment of these tasks.

A robot with the *Attack* task has active interaction with the ball. With this task, the robot is able: i) to block opposite direct shots, ii) to be ready for one-vs-one fights, and iii) to get possession of the ball in case the previous attacker is taken out of the match. In possession of the ball, the robot will try to score either by kicking directly or by dribbling to get a better position for kicking the ball. The robot will also reach the ball and search for the ball in case the robot does not possess it. When searching for the ball, the robot goes first to the place where the ball was last seen. Reaching the ball means to place the robot behind the ball so it can kick or dribble.

The *defender* robot is not supposed to have contact with the ball but to be ready to change its task and approach the ball if necessary.

The task assignment is based on an asynchronous request-and-response system that ensures that there is only one robot actively interacting with the ball. This prohibits, for example, that two robots try to kick the ball simultaneously, which could lead to team self-collisions. The request for a task reassignment depends on the state of the robot and its teammates.

## 4 Conclusions

We are looking forward to participating in RoboCup 2020 in Bordeaux. We want to demonstrate advanced team play strategies in 2 vs. 2 games, which are still under development, and evaluate the performance of our improved deep neural networks used for detections.

## Team Members

Team NimbRo commits to participating in RoboCup 2020 in Bordeaux, France, and to provide a referee knowledgeable of the rules of the Humanoid League. Currently, the NimbRo soccer team consists of the following members:
**Team leader:** Sven Behnke
**Team members:** Hafez Farazi, Grzegorz Ficht, Diego Rodriguez, Dmytro Pavlichenko, Mojtaba Hosseini, Oleg Kosenko, and Marcell Missura.

**Acknowledgements** This research has been supported by the Deutsche Forschungsgemeinschaft (German Research Foundation, DFG) under grant BE 2556/13.

## References

1. Allgeuer, P., Behnke, S.: Fused Angles: A representation of body orientation for balance. In: Int. Conf. on Intelligent Robots and Systems (IROS) (2015)
2. Allgeuer, P., Behnke, S.: Omnidirectional bipedal walking with direct fused angle feedback mechanisms. In: Proceedings of 16th IEEE-RAS Int. Conference on Humanoid Robots (Humanoids). Cancún, Mexico (2016)
3. Allgeuer, P., Farazi, H., Ficht, G., Schreiber, M., Behnke, S.: The igus Humanoid Open Platform: A child-sized 3D printed open-source robot for research. In: Robot Soccer World Cup XX. Springer (2016), lNCS 9776
4. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(12), 2481–2495 (2015)
5. Behnke, S.: Online trajectory generation for omnidirectional biped walking. In: Proceedings of 2006 IEEE Int. Conf. on Robotics and Automation (ICRA) (2006)
6. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Freezeout: Accelerate training by progressively freezing layers. arXiv:1706.04983 (2017)
7. Farazi, H., Allgeuer, P., Behnke, S.: A monocular vision system for playing soccer in low color information environments. In: 10th Workshop on Humanoid Soccer Robots, IEEE-RAS Int. Conference on Humanoid Robots (2015)
8. Farazi, H., Allgeuer, P., Ficht, G., Brandenburger, A., Pavlichenko, D., Schreiber, M., Behnke, S.: RoboCup 2016 Humanoid TeenSize winner NimbRo: Robust visual perception and soccer behaviors. In: RoboCup 2016: Robot Soccer World Cup XX. Springer (2017)

9. Farazi, H., Behnke, S.: Real-time visual tracking and identification for a team of homogeneous humanoid robots. In: RoboCup 2016: Robot World Cup XX. pp. 230–242. Springer (2016)
10. Farazi, H., Behnke, S.: Online visual robot tracking and identification using deep LSTM networks. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017)
11. Ficht, G., Pavlichenko, D., Allgeuer, P., Farazi, H., Rodriguez, D., Brandenburger, A., Kürsch, J., Behnke, S.: Grown-up NimbRo robots winning RoboCup 2017 Humanoid AdultSize soccer competitions. In: RoboCup 2017: Robot Soccer World Cup XXI. Springer (2018)
12. Ficht, G., Farazi, H., Brandenburger, A., Rodriguez, D., Pavlichenko, D., Allgeuer, P., Hosseini, M., Behnke, S.: NimbRo-OP2X: Adult-sized open-source 3D printed humanoid robot. In: 18th IEEE-RAS International Conference on Humanoid Robots (Humanoids) (2018)
13. Missura, M., Behnke, S.: Self-stable omnidirectional walking with compliant joints. In: 8th Workshop on Humanoid Soccer Robots, Humanoids Conference (2013)
14. Niloofar Azizi, Hafez Farazi, Behnke, S.: Location dependency in video prediction. In: International Conference on Artificial Neural Networks (ICANN) (2018)
15. Rodriguez, D., Farazi, H., Allgeuer, P., Pavlichenko, D., Ficht, G., Brandenburger, A., Kürsch, J., Behnke, S.: Advanced soccer skills and team play of RoboCup 2017 TeenSize winner NimbRo. In: RoboCup 2017: Robot Soccer World Cup XXI. Springer (2018)
16. Rodriguez, D., Brandenburger, A., Behnke, S.: Combining Simulations and Real-robot Experiments for Bayesian Optimization of Bipedal Gait Stabilization. RoboCup International Symposium, Montreal, Canada (2018)
17. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer (2015)
18. Schnekenburger, F., Scharffenberg, M., Wülker, M., Hochberg, U., Dorer, K.: Detection and localization of features on a soccer field with feedforward fully convolutional neural networks (FCNN) for the Adult-size humanoid robot Sweaty. In: Proceedings of the 12th Workshop on Humanoid Soccer Robots, IEEE-RAS International Conference on Humanoid Robots (Humanoids). (2017)
19. Smith, L.N.: Cyclical learning rates for training neural networks. In: Applications of Computer Vision (WACV), IEEE Winter Conference on. pp. 464–472 (2017)
20. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems (NIPS). pp. 3320–3328 (2014)