Terrain Classification with Crawling Robot using Long Short-Term Memory Network

Rudolf J. Szadkowski[1] [0000-0003-4075-116X], Jan Drchal[1] [0000-0003-0466-275X], and Jan Faigl[1] [0000-0002-6193-0792]

Czech Technical University in Prague, Technicka 2, 16627 Prague, Czech Republic
{szadkrud, drchajan, faiglj}@fel.cvut.cz

Abstract. Terrain classification is a crucial feature for mobile robots operating across multiple terrains. One way to learn a terrain classifier is to use a stream of labeled proprioceptive data recorded during a terrain traversal. In this paper, we propose a new terrain classifier that combines a feature extraction from a data stream with the long short-term memory (LSTM) network. Features are extracted from the information-sparse data stream by applying a sliding window computing three central moments. The feature sequence is continuously classified by the LSTM network into multiple terrain classes. Furthermore, a modified bagging method is used to deal with a limited and unbalanced training set. In comparison to the previous work on terrain classifiers for a hexapod crawling robot using only servo-drive feedback, the proposed classifier provides continuous classification with the F1 score up to 0.88, and thus provide better results than SVM classifier learned on the same input data.

1 Introduction

Continuous proprioception processing is essential for crawling robots that adapt their locomotion to particular terrain type. In the animal world, a proprioceptive signal carries information about locomotor organs such as muscle stretch or muscle force output [17, 4]. For multi-legged walking robots, the proprioception describes the state of joint or servomotor actuators, and since the state of actuators is correlated with the robot surrounding environment, it is possible to use the proprioception for a local terrain classification [9, 18]. A terrain classifier can be integrated into locomotion control of a hexapod, a six-legged walking robot, to improve the performance [1] such as speed or stability. The robot is controlled in real time, and therefore, the proprioceptive data must be processed continuously to make the terrain classifier synchronous with the locomotion control.

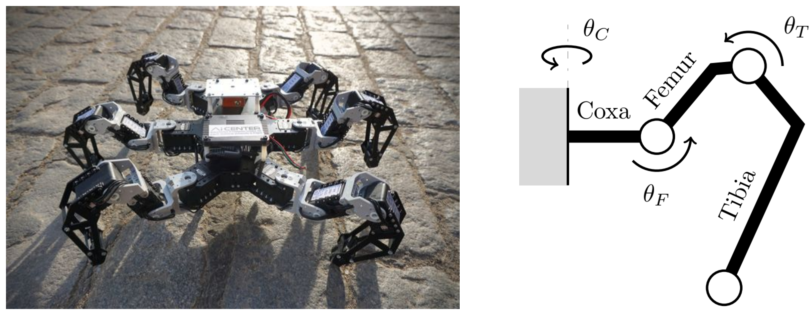
Two types of terrain classification can be distinguished: local and remote [11]. The remote classification relies on ranged exteroceptive sensors, e.g., camera [1] and range sensors such as LiDARs [7, 19]. The local classification relies on proprioception [10] or local exteroception [12], which measures the environment in the close vicinity of the robot body that can be used to select an appropriate motion gait [13]. On the other hand, the primary function of proprioception is to sense the internal state of the body (i.e., muscle stretch pressure or a joint angle) and to participate in the locomotion control. Contrary to local exteroceptive sensors that generate extra costs, the proprioception is usually already on board of multi-legged robots. Therefore proprioceptive signals can be considered as an alternative to the local exteroception for the immediate experience of the robot with the terrain the robot is currently traversing [9, 18].

One of the proprioceptive signals generated by a walking hexapod robot is a sequence of joint angle errors. The joint angle error is a difference between an actual joint angle and desired joint angle which is given by a repetitive locomotion pattern, a gait. In [2], authors classified the terrain using sequences of joint angle errors generated by a simple periodic gait. This simple gait; however, limited the robot to traverse only the flat terrains. To traverse irregular terrains [9] introduces an adaptive gait that adapts the motion to irregularities. Even though the adaptive gait is repetitive, it is not periodic; therefore the adaptive gait cannot be used with classifier [2]. The paper [8] addresses this issue by parsing the gait phases into segments of the same size and then embedded the segments into a feature vector. However, this method relies on prior knowledge about the gait phases, which is not always available. Moreover, SVM-based methods [8, 2] have to wait three gait-cycles to get enough data to produce the feature vector.

We propose to describe the terrain classification as the continuous classification conditioned on a periodic stream of proprioceptive signals. We implemented the continuous classifier as a bagging ensemble [3] combining several Long-Short Term Memory (LSTM) networks [6]. In the ideal case, such a classifier should be trained with a sufficiently large and well-balanced dataset. However, each dataset collection is a costly operation as it requires a complex experimental setup, real robots, and most importantly a human supervisor. Moreover, datasets collected during usual deployments (e.g., exploration) are generally not balanced as it depends on the deployment location. Therefore, in practice, we deal with datasets that are small and unbalanced. We aggregate several LSTM networks into a bagging predictor [3] to address this issue. In particular, we use asymmetric bootstrapping [15] that artificially balances the dataset. We propose a method that exploits the periodic properties of the proprioceptive signal to generate new datasamples, and thus enlarges the dataset. The performance of the proposed predictor is statistically compared with the former SVM-based approach [8]. Regarding the reported results, the proposed method achieves competitive performance while its main benefit is in a continuous prediction.

2 Proprioceptive Signals and Data Collection

The robot classifies the terrain it traverses by processing the stream of proprioceptive signals. We work with the hexapod depicted in Fig. 1(a) which consists of a body and six legs each with three joints connecting body, coxa, femur, and tibia, see Fig. 1(b). When the hexapod traverses a terrain, it moves its joints in a repetitive pattern called a gait. A single repetition of the pattern is called a gait-cycle. A particular gait is defined by a motion pattern, e.g., a robot walking



(a) (b)

Fig. 1. The utilized hexapod and schema of its leg.

with a tripod gait always has at least three legs on the ground in the supporting phase, and three legs are simultaneously moving forward in the swing phase. The gait rules utilized in this paper are conditioned on the terrain interaction, which makes the gait adaptive [9].

2.1 Adaptive Gait

In [9], the authors take advantage of the proprioceptive signals provided by the servomotors to detect terrain irregularities. During a single gait-cycle, each leg goes through four phases: up, forward, down, and support. For each i-th leg and each j-th joint, two variables are monitored: the current angle $\theta_{i,j}^{cur}$ and the desired angle $\theta_{i,j}^{des}$. The joint angle error is defined as the absolute difference between the current and desired angles

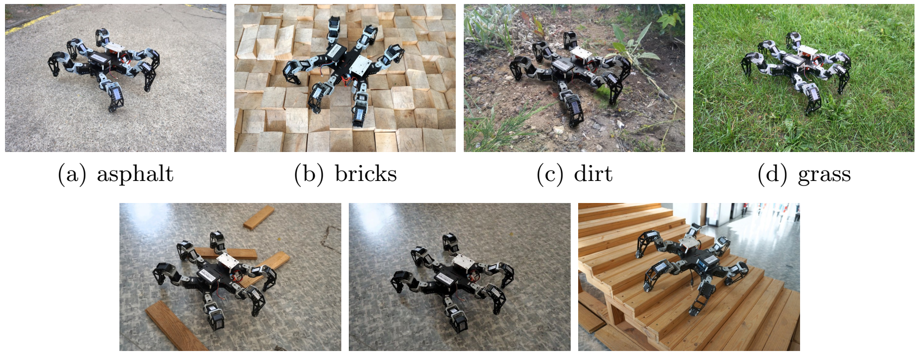
$$\theta_{i,j}^{err} = |\theta_{i,j}^{cur} - \theta_{i,j}^{des}|. \tag{1}$$

During the i-th leg swing-down phase, the error of the body-coxa joint, $\theta_{i,C}^{err}$, is compared with a predefined threshold. If $\theta_{i,C}^{err}$ is above the threshold, it is assumed the deviation is caused by the ground reaction force, and therefore, the motion is stopped and the i-th leg enters into the support phase. Once all moving legs are in the support phase, the body leveling is initiated and move the robot forward. The process is repeated for the next subset of moving legs.

2.2 Data Collection and Preprocessing

The herein proposed approach uses the same data source as in [8] where the SVM classifier processes the angle errors $\theta_{i,j}^{err}$ of the two front legs to classify the terrain. To collect the dataset, we let the hexapod crawl on seven types of terrain: office, asphalt, dirt, bricks, obstacles, stairs, and grass (see Fig. 2).

In each session, the hexapod executes up to ten gait-cycles on a single terrain type. The number of collected gait-cycles for each terrain is shown in Tab. 1. For



(a) asphalt (b) bricks (c) dirt (d) grass

(e) obstacles (f) office (g) stairs

Fig. 2. The hexapod deployed in various terrains for data collection.

each gait-cycle, we recorded the angle errors of the front leg joints, $\boldsymbol{\theta}^{err} \in \mathbb{R}^6$ with the uniform sampling rate. Due to the adaptation to the terrain irregularities, the length of each record of errors may differ. Each gait-cycle record is preprocessed by a sliding window method which computed the mean, standard deviation, and skewness. The width of the window is set to 20 units and the window jumps ahead 5 units. Thus, the preprocessing yielded a sequence of feature vectors $\boldsymbol{x}$, where each feature vector has 18 dimensions (2 legs $\times$ 3 joints $\times$ 3 central moments).

Table 1. Numbers of the sampled gait-cycles and division to train and test sets.

Dataset	Asphalt	Bricks	Dirt	Grass	Obstacles	Office	Stairs
Train set	69	26	56	66	61	77	87
Test set	18	9	15	17	16	20	22
Complete set	87	35	71	83	77	97	109

3 Proposed Terrain Predictor

The proposed terrain predictor is based on the basic LSTM model using the bagging extension to deal with the small and unbalanced data. The addressed classification task can be formalized as follows. Let C be a finite set of terrain classes. Our goal is to find a predictor ϕ^* that predicts a distribution over C for each feature vector $\boldsymbol{x}(m)$ in a continuous feature vector stream. Assuming that at the m-th iteration the distribution is conditioned on the sequence $\boldsymbol{x}^m =$

$(\boldsymbol{x}(m), \boldsymbol{x}(m-1), \ldots, \boldsymbol{x}(1))$, we denote the output of the predictor ϕ^* as

$$\boldsymbol{y}(m) = (P(C = c_1|\boldsymbol{x}^m), P(C = c_2|\boldsymbol{x}^m), \ldots, P(C = c_{|C|}|\boldsymbol{x}^m)), \tag{2}$$

where $P(C = c_i|\boldsymbol{x}^m)$ is probability that the class at the m-th step is $c_i \in C$. The continuous prediction over the sequence $(\boldsymbol{x}(m), \boldsymbol{x}(m-1), \ldots, \boldsymbol{x}(1))$ then yields a sequence of the probability distributions $(\boldsymbol{y}(m), \boldsymbol{y}(m-1), \ldots, \boldsymbol{y}(1))$.

The terrain prediction (2) can be considered as the sequence-to-sequence problem where the input sequence is mapped to the output sequence. We propose to approximate ϕ^* by the neural network ϕ composed of a single LSTM hidden layer (see [6] for equations) with the softmax output layer. In the training phase, each i-th training pair

$$((\boldsymbol{x}_i(M_i), \boldsymbol{x}_i(M_i - 1), \ldots, \boldsymbol{x}_i(1)), d_i) \tag{3}$$

is presented to the neural network ϕ, where M_i is the length of the training sequence. The desired class d_i is time-invariant because the terrain class does not change during the training sequence. For each feature vector $\boldsymbol{x}_i(m)$, we get the output $\boldsymbol{y}_i(m)$ that is compared with the desired class d_i using the loss function $\mathcal{L}(\boldsymbol{y}_i(m), d_i)$. We followed a common practice with neural network classifiers, and we chose the cross-entropy error as the loss function. The loss of the whole i-th training sequence is then evaluated as

$$\mathcal{L}(\boldsymbol{y}_i, d_i) = \sum_{m=M_{min}}^{M_i} \mathcal{L}(\boldsymbol{y}_i(m), d_i), \tag{4}$$

where M_{min} denotes the offset of the first feature vector in the sequence that is being evaluated. Preliminary experiments showed that it is better to leave several initial samples unevaluated. The length of the i-th sequence M_i determines how much information about the terrain d_i is provided to the predictor.

The problem of small and unbalanced dataset collected by the robot is evident from Tab. 1 and it is addressed by implementation of the terrain predictor as a bagging ensemble [3] with a modified bootstrapping method. The bagging ensemble is denoted as

$$\phi_B(\boldsymbol{x}) = \frac{\sum_{j=1}^{S} \phi(\boldsymbol{x}; D^j)}{S}, \tag{5}$$

where D^j is the j-th bootstrap dataset, S is the number of the bootstrap datasets, and $\phi(\boldsymbol{x}; D^j)$ is the output of the neural network trained on D^j. The bootstrap datasets are usually generated by taking N random samples with the replacement from the source dataset D. The distribution of the bootstrap datasets then approximates the probability distribution of D [3]. However, in our case, this is undesirable because the source dataset D is unbalanced. Therefore, we propose the modified bootstrapping method described in Algorithm 1. This algorithm uses asymmetric bootstrapping which balances the bootstrap dataset [15]. Then the algorithm creates new samples by combining randomly selected gait-cycle sequences. Note, that by using this random combination we assume that the gait-cycles from the same terrain are independent. After being trained, the proposed predictor does not need to parse the input stream into gait-cycles, i.e., the predictor can work without any knowledge of the gait implementation.

Algorithm 1 Bootstrap dataset generator

Input C: classes; G_i: set of single gait-cycles for class $i \in C$;
L: number of gait-cycles in one sequence; N: size of the bootstrap dataset.
Output D': bootstrap dataset containing $(sequence, class)$ training pairs.

for N times **do**
  $class \leftarrow random\_choice(C)$
  $sequence \leftarrow (\varnothing)$
  for L times **do**
    $gaitcycle \leftarrow random\_choice(G_{class})$
    $sequence.concatenate(gaitcycle)$
  end for
  $trainpair \leftarrow (sequence, class)$
  $D'.add(trainpair)$
end for

4 Experiments

The dataset collected using the method described in Sec. 2.2 is divided into a training dataset and a testing dataset (see Tab. 1), the latter is used only for the evaluation. The Algorithm 1 generates bootstrap datasets with $N = 1000$ training pairs, and each training sequence contains $L = 3$ gait-cycles because it should contain enough information to classify the terrain [8]. The average length of the training sequence is 73 and M_{min} is set to 50. We generate 30 bootstrap datasets for 30 neural networks, where each network consists of 20 hidden LSTM units with forget gate [5], the input layer has 18 units, and the output layer has seven units corresponding to the particular terrain classes.

We use the `rmsprop` [16] with the learning rate set to 0.01 and decay rate α set to 0.99 to backpropagate the error. During one epoch, each training sequence is forward-passed and backpropagated. Therefore, the learning algorithm performs 1000 backpropagation iterations per one epoch, and each network is trained on 100 epochs. Finally, all 30 trained networks are aggregated into the bagging predictor. For the evaluation, we generated testing sequences composed of four gait-cycles instead of three gait-cycles that are used during training, because we aim to study the temporal generalization of the networks.

Two examples of how the terrain distribution prediction changes in time are shown in Fig. 3. The performed evolution of the prediction accuracy for each type of the classified terrain is shown in Fig. 4. Based on the results, it seems

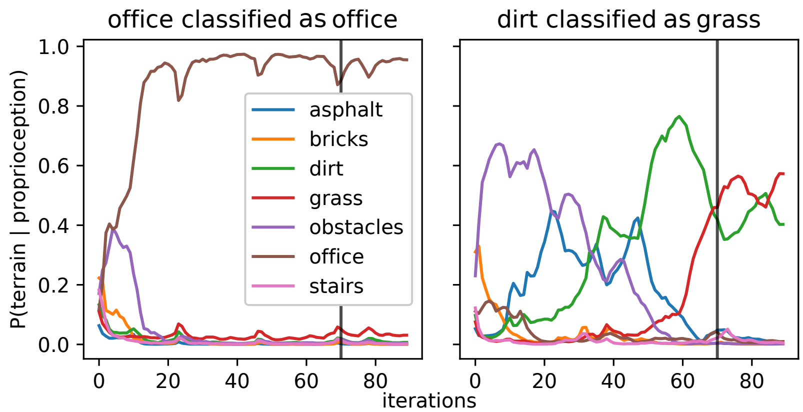


Fig. 3. Example of terrain probability distribution changes generated by the proposed predictor. On the left, the office (brown) is correctly predicted with high certainty, after 20 iterations. On the right, dirt (green) is mispredicted as an obstacle (violet) and then as a grass (red).

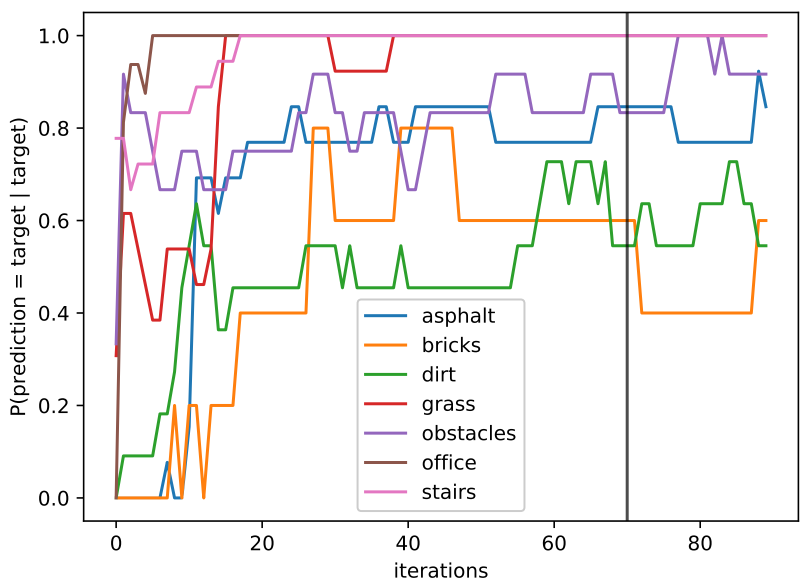


Fig. 4. Evolution of the prediction accuracy for each terrain type. Office floor, stairs, and grass terrain types are classified at almost 100% at the 40th iteration. The accuracy of each terrain settles up around the 70th iteration (marked by the vertical line), which roughly corresponds to the end of the 3rd gait-cycle.

that for each terrain, the accuracy settles up at a different iteration step, and thus each terrain requires a different amount of the proprioceptive data to be classified with high confidence. Another observation is that after 40 iterations, which roughly corresponds to one and half a single gait-cycle, the prediction accuracy of the grass, office, and stairs terrains is almost perfect. The confusion matrix evaluated on the 70th iteration can be found in Tab. 2.

Table 2. Confusion matrix evaluated at the end of the 70th iteration (about the end of 3rd gait-cycle).

	Asphalt	Bricks	Dirt	Grass	Obstacles	Office	Stairs
Asphalt	**12**	0	2	0	0	0	0
Bricks	0	**3**	0	0	0	0	2
Dirt	0	0	**6**	5	0	0	0
Grass	0	0	0	**13**	0	0	0
Obstacles	0	0	2	0	**10**	0	0
Office	0	0	0	0	0	**16**	0
Stairs	0	0	0	0	0	0	**18**

Finally, we compared the bagging ensemble with the SVM classifier utilized in [8]. The comparison is not straightforward because both models are qualitatively different. Our ensemble predicts continuously through iterations as can be seen in Fig. 3 contrary to the SVM classifier [8] that relies on the well defined gait-cycle phases. Therefore, we also consider an uninformed variant of the approach [8] where the feature vector does not contain information about gait-cycle phases. The comparison is shown in Tab. 3 where we use the weighted F1 score [14] because the testing dataset is unbalanced.

Table 3. Predictor comparison using the weighted F1 score [14]. All the predictors are trained and tested using the same training set and test set except for the SVM classifier [8] which uses information about gait-cycle phases. The predictors are considered for the sequences of different lengths up to four gait-cycles.

Predictor	Gait-cycles			
	1	2	3	4
Bagging predictor	**0.83**	**0.86**	0.87	0.88
SVM uninformed	0.63	0.75	0.79	0.77
Single LSTM predictor	0.66	0.78	0.83	0.82
SVM [8]	0.54	0.78	**0.88**	**0.90**

Discussion – The results in Fig. 4 indicate that the prediction accuracy is almost perfect for office, dirt, and stairs terrains. We hypothesize that it is because these three terrains are mutually well distinguishable. From the results in Tab. 2 we can see that the classifier confuses intuitively similar terrains. An example of such confusion can be seen in Fig. 3. From Fig. 4, it is also evident that for the classification, each terrain needs a different number of iterations. This can be exploited by classifiers that process every feature vector continuously. In that regard, the proposed continuous processing of the proprioceptive data adds a qualitative advantage over non-continuous approaches [2, 8].

5 Conclusion

In this paper, we report on the proposed LSTM based terrain predictor suitable for a hexapod crawling robot using proprioceptive signals to process a stream of the joint angle errors generated during crawling irregular terrains by the adaptive locomotion. Due to a small and imbalanced dataset, the basic LSTM methods are not directly applicable. Therefore, we propose to wrap multiple LSTM predictors into a bagging ensemble using a modified bootstrapping algorithm to deal with the class imbalance. The proposed modification takes advantage of the periodicity of the input stream to enlarge the dataset artificially. The resulted bagging predictor has been statistically compared with the existing SVM-based predictor utilized in the previous work on the terrain classification using a real hexapod crawling robot. The main advantage of the proposed solution is that, unlike the SVM-based predictor, it can provide prediction each iteration step. The reported results demonstrate that different terrains need a different amount of the input information to get prediction with high confidence. Therefore the proposed formulation of the terrain classification task as the sequence-to-sequence problem seems to be suitable for processing stream of proprioceptive signals.

The main shortcoming of the terrain classification is that it depends on the gait used for the training. Different gaits have different properties such as the servomotor load, and thus the particular gait influences the patterns of the proprioceptive signals. The proposed classifier is designed with the intention to support the locomotion controller, and therefore, we plan to address the influence of the gait to the classification and thus improve the transferability to different gait types in our future work.

Acknowledgments – The presented work has been supported by the Czech Science Foundation (GAČR) under research project No. 18-18858S. The support of grant No. SGS16/235/OHK3/3T/13 to Rudolf Szadkowski is also gratefully acknowledged. Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures" (CESNET LM2015042), is greatly appreciated.

References

1. Bartoszyk, S., Kasprzak, P., Belter, D.: Terrain-aware motion planning for a walking robot. In: 2017 11th International Workshop on Robot Motion and Control

(RoMoCo). pp. 29–34 (2017)
2. Best, G., Moghadam, P., Kottege, N., Kleeman, L.: Terrain classification using a hexapod robot. In: Australasian Conference on Robotics and Automation (2013)
3. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (Aug 1996)
4. Frigon, A., Rossignol, S.: Experiments and models of sensorimotor interactions during locomotion. Biological Cybernetics **95**(6), 607 (Nov 2006)
5. Gers, F.: Long short-term memory in recurrent neural networks. Unpublished PhD dissertation, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland (2001)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**, 1735–1780 (12 1997)
7. McDaniel, M.W., Nishihata, T., Brooks, C.A., Salesses, P., Iagnemma, K.: Terrain classification and identification of tree stems using groundbased lidar. Journal of Field Robotics **29**(6), 891–910 (2012)
8. Mrva, J., Faigl, J.: Feature extraction for terrain classification with crawling robots. In: Information Technologies - Applications and Theory. vol. 1422, pp. 179–185 (2015)
9. Mrva, J., Faigl, J.: Tactile sensing with servo drives feedback only for blind hexapod walking robot. In: 10th International Workshop on Robot Motion and Control (RoMoCo). pp. 240–245 (2015)
10. Ojeda, L., Borenstein, J., Witus, G., Karlsen, R.: Terrain characterization and classification with a mobile robot. Journal of Field Robotics **23**(2), 103–122 (2006)
11. Otsu, K., Ono, M., Fuchs, T.J., Baldwin, I., Kubota, T.: Autonomous terrain classification with co- and self-training approach. IEEE Robotics and Automation Letters **1**(2), 814–819 (July 2016)
12. Otte, S., Weiss, C., Scherer, T., Zell, A.: Recurrent neural networks for fast and robust vibration-based ground classification on mobile robots. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 5603–5608 (2016)
13. Rebula, J.R., Neuhaus, P.D., Bonnlander, B.V., Johnson, M.J., Pratt, J.E.: A controller for the littledog quadruped walking on rough terrain. In: IEEE International Conference on Robotics and Automation (ICRA). pp. 1467–1473 (2007)
14. Sasaki, Y., et al.: The truth of the f-measure. Teach Tutor mater **1**(5) (2007)
15. Tao, D., Tang, X., Li, X., Wu, X.: Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(7), 1088–1099 (2006)
16. Tieleman, T., Hinton, G.: Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural networks for machine learning **4**(2), 26–31 (2012)
17. Tóth, T.I., Knops, S., Daun-Gruhn, S.: A neuromechanical model explaining forward and backward stepping in the stick insect. Journal of Neurophysiology **107**(12), 3267–3280 (2012)
18. Walas, K., Kanoulas, D., Kryczka, P.: Terrain classification and locomotion parameters adaptation for humanoid robots using force/torque sensing. In: IEEE-RAS 16th International Conference on Humanoid Robots. pp. 133–140 (2016)
19. Walas, K., Nowicki, M.: Terrain classification using laser range finder. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 5003–5009 (2014)