2020 4th International Conference on Automation, Control and Robots

Transfer of Inter-Robotic Inductive Classifier

Rudolf Szadkowski, Miloš Prágr, Jan Faigl
Faculty of Electrical Engineering
Czech Technical University in Prague
Technicka 2, 166 27, Prague, Czech Republic
e-mail: {szadkrud|pragrmi1|faiglj}@fel.cvut.cz

Abstract—**In multi-robot deployments, the robots need to share and integrate their own experience and perform transfer learning. Under the assumption that the robots have the same morphology and carry equivalent sensory equipment, the problem of transfer learning can be considered incremental learning. Thus, the transfer learning problem inherits the challenges of incremental learning, such as catastrophic forgetting and concept drift. In catastrophic forgetting, the model abruptly forgets the previously learned knowledge during the learning process. The concept drift arises with different experiences between consecutively sampled models. However, state-of-the-art robotic transfer learning approaches do not address both challenges at once. In this paper, we propose to use an incremental classifier on a transfer learning problem. The feasibility of the proposed approach is demonstrated in a real deployment. The robot consistently merges two classifiers learned on two different tasks into a classifier that performs well on both tasks.**

Keywords-component; multi-robotics; incremental learning; inductive transfer learning; classification

I. Introduction

In multi-robot systems, there is an important step the robots need to take to become a fully integrated collective. It is the ability to transfer and integrate experience gained by each other individual robot. The goal of the experience transfer is to improve the robot model by integrating the experience of other robots that operate in different environments or perform various tasks [1–3]. The problem of experience transfer is known in machine learning as transfer learning, and it is still an open problem of integrating a source model (or models) into a target model, and thus increase the performance of the target model [4]. In this work, we focus on transfer learning between robots that share the same morphology and sensory equipment and learn to classify terrain.

The homogeneous robots experience the environment using the same sensory equipment, and therefore, they perceive the environment similarly; however, space and time where and when the experience is drawn differ. Transferring models learned on experiences drawn from the same distribution but under different tasks (conditions) is a well-defined problem known as inductive transfer learning [5]. During inductive transfer learning, a source model, which is trained during a particular task, is integrated into the target model that is then utilized for a similar testing task; the source model should improve the performance of the target model on the testing task. The problem of integrating one model into another can be addressed as incremental learning, where the model is transferred from the past into the present.

The herein addressed inductive transfer learning inherits two main challenges of incremental learning: catastrophic forgetting, when the target model abruptly forgets its previously learned knowledge; and concept drift when the underlying concept is different in the source and target models [6]. In the current state-of-the-art on inductive transfer learning, to the best of the authors' knowledge, no method addresses both challenges simultaneously.

In our previous work [7], we propose a memory-replay-based incremental learning algorithm ENSGENDEL, which is designed to resist catastrophic forgetting and adapt to concept drift. In this paper, we propose to use the incremental learning algorithm ENSGENDEL for the inductive transfer learning problem. Moreover, we adapt the minimal evaluation scenarios for incremental algorithms [7] for inductive transfer learning in the designed robotic deployment visualized in Fig. 1. Inductive transfer learning is deployed on real hexapod walking robots and evaluated on the adapted minimal evaluation scenarios, demonstrating the feasibility of the proposed solution.

The rest of the paper is organized as follows. The related work on terrain classification and transfer learning in robotics are overviewed in Section II. The problem statement and the method are described in Section III and Section IV, respectively. The presented transfer learning algorithm deployed on a real walking hexapod robot is presented in Section V, and the results of the experimental deployment are discussed in Section VI. Finally, the paper is concluded in Section VII.

II. Related

The transfer learning proposed in this paper is focused on the transfer of classification models. Specifically, we consider terrain classification utilized in robot navigation through operational environments with various terrain types, where a selection of the suitable terrain and control might improve the robot locomotion. Therefore, a brief overview of terrain classification is presented to provide the necessary background to the addressed learning problem.

978-1-7281-9207-9/20/$31.00 ©2020 IEEE

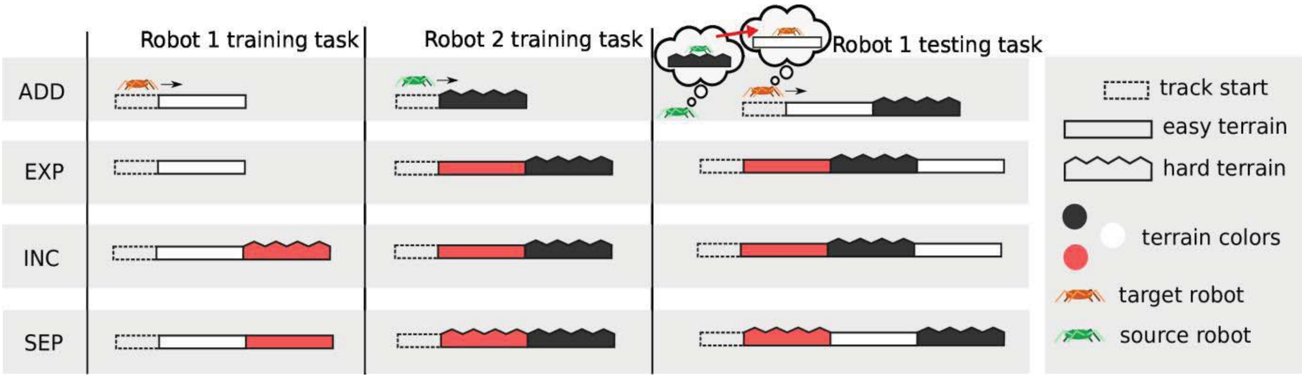

Figure 1. The minimal evaluation scenarios adapted for inductive transfer learning, where two robots learn to predict hard/easy terrain based on the color of a fabric laid down on the terrain. In each of the four scenarios, there are three tasks (tracks): in the first and second tasks, the robots train on their respective tasks; then a transfer learning is executed where the model of the second (source) robot is merged into the first (target) robot model, which is then evaluated in the third testing task. Each scenario tests a specific aspect of inductive transfer learning: the addition scenario (ADD) tests the ability to add a new class into the classifier, and the expansion scenario (EXP) tests the ability to expand an existing class, both without forgetting; the inclusion scenario (INC) tests the ability to adapt to concept drift, and finally the separation scenario (SEP) tests the ability to unlearn without forgetting [7].

The authors of [8] use terrain geometry to discriminate impassable parts of the environment in rough terrain planning. Beyond passability prediction, the terrain can be described by its type, such as sand or concrete, which can be predicted by the model presented in [9], where Support Vector Machines (SVM) and Random Trees classify terrains using color and elevations descriptors. A similar problem is addressed in [10], where the authors combine a vibration-based SVM classifier with an SVM laser-based classifier and a Gaussian mixture. Moreover, we investigate terrain-traversal time-series segment classification in [11] as a part of our effort on incremental terrain learning for autonomous mobile robots [12, 13]. The terrain learning can speed up multi-robot deployment, where multiple robots are exploring the terrain in parallel. Sharing their individually learned models through transfer learning can thus support better exploitation of the acquired knowledge about the operational environment, which motivates the presented approach on inductive learning.

In transfer learning scenarios, the learner applies to its current task a model learned on the previously sampled tasks [4, 5]. In robotics, transfer learning commonly entails scenarios with multiple robots. Two heterogeneous robots are deployed in [1] and apply the transfer learning for a place recognition scenario. Another scenario with heterogeneous robot deployment is presented in [2], where the employed algorithm represents measured data from different sensors in a learned abstract feature space. The therein proposed approach is extended in [14] to provide distributed robot team control on missions such as search-and-rescue. A prediction of the traversal cost over the terrain observed from an aerial scan using a model learned by a ground crawler is proposed in [15]. In that scenario, the transfer learners aim for knowledge transfer between heterogeneous robots. On the other hand, transfer learning can be transformed into incremental learning for homogeneous robots.

Incremental learning concerns learning scenarios where the training data arrive one-by-one and where the learning algorithm is constrained by limited memory resources [6]. Existing incremental learning algorithms range from incremental versions of the SVM [16] to self-organizing networks such as [17]. Regarding the particular learning algorithm employed in this paper, the proposed approach is built on the results of [7], where we propose an incrementally learnable classifier. Besides, we leverage on the proposed scenarios to evaluate the ability of incremental learning to be robust against catastrophic forgetting and concept drift. The relation between the utilized incremental learning approach and the addressed inductive learning is described in the following section.

III. Problem Statement and Analysis

This paper concerns a scenario where multiple homogeneous robots are learning to classify terrain, share their models, and improve their performance using inductive transfer learning. The goal of the classifier transfer learning is to improve the target classifier by merging it with the source classifier, where target and source classifiers are trained on different tasks. In the transfer learning problem, each i-th model is trained in specific domain and task [5]. The domain can be formally defined as $\Delta^i = (\chi, P^i(X))$, where χ is a feature space and $P^i(X)$ is the marginal distribution over the finite discrete feature set $X \subset \chi$. The task $T^i = (Y, F^i(.))$ is defined by the label space Y and the model $F^i\!: \chi \rightarrow Y$ that is trained by the dataset $D^i = \{(x, y)\}_{x \in X^i}$, where $y \in Y$ are the observed labels.

Each model is learned by a different robot deployed within the same environment but at a different location or time. Since the robots are also homogeneous, i.e., have the same body morphology and sensory equipment, we assume that the robots learn within the common domain $\Delta = (\chi, P(X))$. However, the individual tasks are different since the robots operate at different locations or time, and thus they are trained on different tasks $T^i \neq T^j$. In particular, the robots learn models $F^i(.) \neq F^j(.)$, since they collect the datasets $D^i \neq D^j$. Transfer learning where the models are trained in the common domain Δ, but under different tasks, $T^i \neq T^j$ is called inductive transfer learning.

The description of the inductive transfer learning where the model F^i is trained during the task T^i with the dataset D^i is equivalent to the description of incremental learning presented in [7], where instead of multiple robots (agents) merging their models, one agent merges models from the past into the present. Thus, the only difference between the inductive transfer learning and incremental learning is the semantic interpretation of the index i. In transfer learning, the index i is the robot index, while in incremental learning, the index i represents the time-step index. Therefore, for inductive transfer learning, we can use the same method as for incremental learning.

IV. Method

In the presented inter-robotic classifier, we focus on the simplest case of inductive transfer learning where two robots $i \in \{1, 2\}$ train the classifiers $F^i: \chi \rightarrow \{L1, L2\}$. We adapt the results presented in [7] to learn the model and evaluate the result: the basic scenarios of incremental classification, and the incremental learning algorithm ENSGENDEL described in Alg. 1.

Algorithm 1: The ENSGENDEL update.

Variables: During a task, the model F is given a dataset $D = \{(\boldsymbol{x}^1, y^1), (\boldsymbol{x}^2, y^2), \ldots\}$, where $\boldsymbol{x} \in X$ and $y \in \{L1, L2\}$. The dataset is used to train model $F = \{e_{L1}, e_{L2}, d_{L1}, d_{L2}, f_{L1}, f_{L2}\}$, which is implemented as a set of neural networks. The networks $e_l: \chi \rightarrow Z$, $d_l: Z \rightarrow \chi$ are the encoder and decoder of the autoencoder that is used to generate the features of the l-th class. The network $f_l: \chi \rightarrow [0, 1]$ is a discriminator network trained to distinguish whether the feature $\boldsymbol{x}$ belongs to the l-th class. The parameters N, M, ε, and θ are the number of generated samples, maximum epoch, deletion neighborhood radius, and acceptance threshold, respectively.

function UPDATE(D, F):
  for l **in** $\{L1, L2\}$ **do**
    $A_l \leftarrow \{\boldsymbol{x} | y = l; (x, y) \in D\}$
    $S \leftarrow \{\boldsymbol{s} \in \text{uniform\_sample}([-1, 1]^{\dim(Z)})\}_N$
    $B_l \leftarrow \{d_l(\boldsymbol{s}) \mid f_l(d_l(\boldsymbol{s})) > 0.9; \boldsymbol{s} \in S\}$
  end for
  $C_{L1} \leftarrow A_{L1} \cup \{\boldsymbol{b} \in B_{L1} | \forall \boldsymbol{a} \in A_{L2}: \|\boldsymbol{b} - \boldsymbol{a}\| > \varepsilon\}$
  $C_{L2} \leftarrow A_{L2} \cup \{\boldsymbol{b} \in B_{L2} | \forall \boldsymbol{a} \in A_{L1}: \|\boldsymbol{b} - \boldsymbol{a}\| > \varepsilon\}$
  for (l, k) **in** $\{(L1, L2), (L2, L1)\}$ **do**
    for M **times if** $\exists \boldsymbol{x} \in C_l: \|\boldsymbol{x} - d_l(e_l(\boldsymbol{x}))\| > \theta$ **do**
      $J^f_l \leftarrow \sum_{x \in Cl} \ln(f_l(\boldsymbol{x})) + \sum_{x \in Ck} \ln(1 - f_l(\boldsymbol{x}))$
      $J^{ed}_l \leftarrow \sum_{x \in Cl} \|\boldsymbol{x} - d_l(e_l(\boldsymbol{x}))\|$
      $J_l \leftarrow (|C_l| + |C_k|)^{-1} J^f_l \; - |C_l|^{-1} J^{ed}_l$
      $e_l, d_l, f_l \leftarrow \text{optimize}(J_l, e_l, d_l, f_l)$
    end for
  end for
  return $\{e_{L1}, e_{L2}, d_{L1}, d_{L2}, f_{L1}, f_{L2}\}$
end function

Algorithm 2: Dataset extraction from ENSGENDEL model.

Variables: For a given model $F = \{e_{L1}, e_{L2}, d_{L1}, d_{L2}, f_{L1}, f_{L2}\}$ (learned by the update algorithm described in Alg. 1) the extract algorithm generates a dataset $D = \{(\boldsymbol{x}, y)\}$ of the maximum size $2N$.

function EXTRACT(F):
  for l **in** $\{L1, L2\}$ **do**
    $A_l \leftarrow \{\boldsymbol{x} \mid y = l; (x, y) \in D\}$
    $S \leftarrow \{\boldsymbol{s} \in \text{uniform\_sample}([-1, 1]^{\dim(Z)})\}_N$
    $B_l \leftarrow \{d_l(\boldsymbol{s}) \mid f_l(d_l(\boldsymbol{s})) > 0.9; \boldsymbol{s} \in S\}$
    $D_l \leftarrow \{(\boldsymbol{x}, l) \mid \boldsymbol{x} \in B_l\}$
  end for
  return $\{D_{L1} \cup D_{L2}\}$
end function

In the basic scenarios, the algorithm is trained in two time-steps. The scenarios also provide an evaluation methodology for observing how the incremental learning algorithm handles catastrophic forgetting and concept drift. In the herein presented robotic inductive transfer learning deployment, two hexapod walking robots traverse over terrains, considered hard $L1$ or easy $L2$ to traverse. The robot i trains the classifier F^i, which labels the terrain as hard or easy based on the terrain mean RGB color. The target (first) robot then merges its classifier F^1 with the transferred classifier F^2 from the source (second) robot, creating the merged classifier $\hat{F}^1$. The target robot then traverses the testing track, which combines both the target and source robot tasks. The used four basic scenarios are described in Fig. 1.

The employed incremental learning algorithm ENSGENDEL, which is adapted for transfer learning, comprises an ensemble of generative discriminator neural networks with the ability to delete (untrain) the previously trained knowledge. The model F^i predicts the label with its discriminator network $F^i(x) = \text{argmax}_{l \in Y} f^i_l(x)$. The transfer learning is then implemented as a target robot merging its ENSGENDEL model F^1 with the model of the source robot F^1, where the source model F^2 is used to generate a labeled dataset, see Alg. 2. It then trains the target model F^1, creating the new merged model $\hat{F}^1$.

In each scenario described in Fig. 1, the target robot traverses the first track and collects the dataset D^1. The dataset is utilized to update the target robot model $F^1 \leftarrow UPDATE(D^1, F^1)$ (see Alg. 1). The source robot traverses the second track and updates the source robot model $F^2 \leftarrow UPDATE(D^2, F^2)$. Then, the source robot transfers its model into the target robot, which merges the source model with its own target model $\hat{F}^1 \leftarrow UPDATE(EXTRACT(F^2), F^1)$, see Alg. 2 for the EXTRACT definition.

It is assumed that the merged model $\hat{F}^1$ performs better on the third track of the scenario than the original model F^1, which is verified in the experimental deployment described in the following section.

V. Results

The proposed inductive transfer learning algorithm has been deployed on the real hexapod walking robot shown in Fig. 2. The hexapod robot performed the four scenarios described in Fig. 1. During each scenario, the robot is deployed on the respective track to traverse terrain that appears either red, white, or black, as it is illustrated in Fig. 2b, and which is either hard or easy to traverse. The terrain appearance is described using the RGB color feature $x \in \chi = [0, 1]^3$, and the hard and easy terrain types correspond to the labels $L1$ and $L2$, respectively. The layer sizes of the ENSGENDEL model discriminator f and the auto-encoder generator $d \circ e$ are 3-30-30-15-2 and 3-30-30-15-2-15-30-30-3, respectively. All the layers are composed of rectified linear units (ReLU). We use the Adam optimization algorithm [19] to train the neural networks with the learning rate set to 0.001. The hyperparameters of the update and extract algorithms (Alg. 1 and Alg. 2, respectively) are empirically set to $\theta = 0.002$, $N = 200$, $M = 10$, and $\varepsilon = 0.1$.

(a) (b)

Figure 2. (a) The utilized hexapod walking robot [18] with the Intel RealSense T265 and D435 cameras, providing localization and visual perception with depth information, respectively. (b) Illustration of the robotic setup.

TABLE I. Accuracies of Models with ($\hat{F}^1$) and without (F^1) Transfer Learning. Accuracies are Evaluated on the Testing Track

Model	ADD	EXP	INC	SEP
F_s^1	0.88	0.46	0.39	0.57
$\hat{F}_s^1$	1.00	0.87	0.76	0.78

The features are labeled by a supervisor who remotely controls the robot. The labels are associated with terrain color using the robot localization provided by the Intel RealSense T265, thus creating the datasets D^i_s. The labels are shown at the top of the plots in Fig. 3. In each s-th scenario, the ENSGENDEL models are updated using Alg. 1 $F_s^1 \leftarrow UPDATE(D_s^1, F_s^1)$, $F_s^2 \leftarrow UPDATE(D_s^2, F_s^2)$, and the transfer learning is performed by merging the source model F_s^2 into the target model F_s^1, creating the merged model $\hat{F}_s^1 \leftarrow UPDATE(EXTRACT(F_s^2), F_s^1)$. Then, the merged model $\hat{F}_s^1$ predicts the terrain class on the testing track. The predictions are compared to manually created ground truth. The test track ground truth is used to evaluate the accuracy of the models before F^1. Accuracies of both models with and without transferred knowledge, F^1 and $\hat{F}^1$ respectively, are compared in Table I, where it can be observed that the classifier performance improves after the transfer learning.

The terrain color perceived by RGB-D camera (the Intel RealSense D435) is visualized at the bottom of the individual plots in Fig. 3. The clusters of the perceived RGB features are shown in Fig. 4.

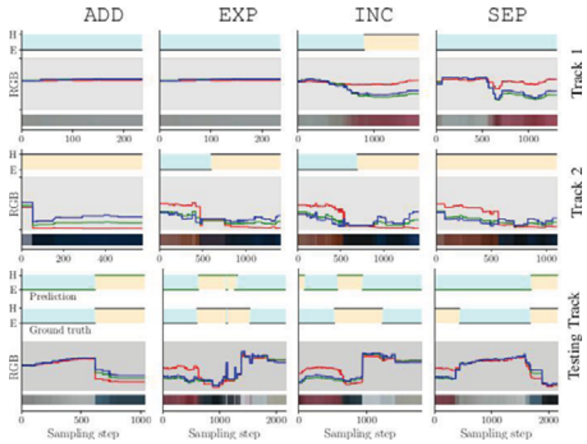

Figure 3. The sequence of RGB features and the corresponding labels captured by the hexapod walking robot traversing the track. Each column and row correspond to a scenario and its respective track. On top of each track plot, there are visualized easy (blue, labeled as E) and hard (orange, labeled as H) terrain labels corresponding to the RGB features visualized at the bottom of the plot as RGB values and color time series. The labels on the first and second tracks are given by the supervisor who remotely controls the robot. There are two label sources on the test track: the prediction provided by the merged model, and the ground truth.

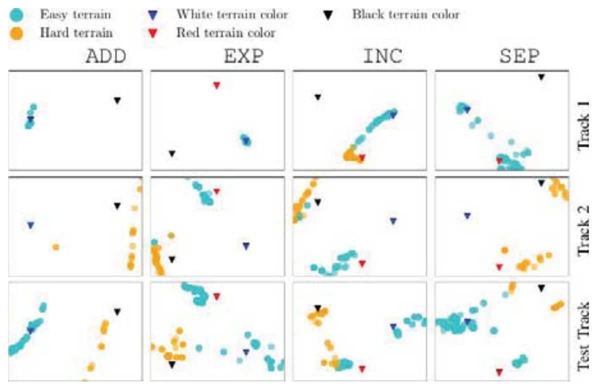

Figure 4. RGB features collected during each scenario projected into a 2D plane using principal component analysis. The projected features are colored by their respective labels, which are given by the supervisor for the first and second tracks, or by the merged model for the test tracks. Triangles mark means of the colors used in the experiment.

VI. Discussion

Each of the four deployment scenarios assesses different aspects of transfer learning. The ADD scenario tests whether the target model is able to integrate a new unknown class from the source model without forgetting the previously learned ones. As can be observed in Fig. 4, in the ADD test track, the predictor assigns labels for both classes. The neighborhoods of $L1$ (easy terrain) features in the test track and in the first track correspond to each other, as both are on the left side; likewise, the neighborhoods of $L2$ (hard terrain) also correspond to each other. A similar result can be observed in the EXP scenario, which tests the ability to expand an existing class, the $L1$ class in this case, without catastrophic forgetting. While the ADD and EXP scenarios evaluate the handling of catastrophic forgetting, the INC and

SEP scenarios specialize in concept drift, which in transfer learning manifests as conflicting source and target models. Such conflict can be observed in Fig. 4 for the INC scenario, where the feature neighborhoods of the first track *L2* and the second track *L1* overlap.

In the presented work, we assume that the source model learned on the second track is the current truth, and thus, as can be observed in the INC test track, the merged model learns to label the features at the bottom as *L1*, which corresponds to the source model labeling. However, in the SEP scenario, it can be observed from Fig. 3 and Fig. 4 that the model fails to unlearn the conflicting feature labels.

Overall, the transfer learning improves the classification accuracy as it can be seen in Table I. Thus, we can conclude that the same approach can solve the inductive transfer and incremental learning. Note, the herein presented scenario deployment is artificially setup using distinct feature clusters so that the transfer learning can be intuitively demonstrated. In our future work, we aim to deploy the presented algorithm in more realistic homogeneous multi-robot deployments to combine both incremental and transfer learning. This paper assumes that the perfect mapping between terrain and traversability property is the same for both homogenous robots. Future research should examine the case of heterogeneous robots, where the terrain traversability can be different for each robot.

VII. CONCLUSION

In this paper, we show that transfer learning for homogeneous robots deployed in a similar environment is equivalent to incremental learning with one robot. The equivalence between incremental and transfer learning is demonstrated by adapting the incremental learning algorithm ENSGENDEL for transfer learning. The adapted ENSGENDEL algorithm is deployed in four robotic scenarios, where a real hexapod walking robot is trained to recognize two types of terrain based on its color. The trained classifiers are then merged within the transfer learning framework producing the merged classifier, which outperforms the baseline classifier without transfer learning. Therefore, we conclude that in the context of deploying multiple homogeneous robots, it is possible to solve both incremental and transfer learning by the same algorithm.

ACKNOWLEDGMENT

The presented work has been supported by the Czech Science Foundation (GAČR) under research project 18-18858S. The support under grant No. SGS19/176/OHK3/3T/13 to Rudolf Szadkowski and Miloš Prágr is also gratefully acknowledged.

REFERENCES

[1] Jie Luo, Andrzej Pronobis, and Barbara Caputo. SVM-based transfer of visual knowledge across robotic platforms. In International Conference on Computer Vision Systems (ICVS), 2007.

[2] Zsolt Kira. Inter-robot transfer learning for perceptual classification. In International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pages 13–20, 2010.

[3] Gabriele Costante, Thomas A. Ciarfuglia, Paolo Valigi, and Elisa Ricci. Transferring knowledge across robots: A risk sensitive approach. Robotics and Autonomous Systems, 65:1–14, 2015.

[4] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In International Conference on Artificial Neural Networks (ICANN), volume 11141 of Lecture Notes in Computer Science, pages 270–279, 2018.

[5] S. J. Pan and Q. Yang. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2010.

[6] Alexander Gepperth and Barbara Hammer. Incremental learning algorithms and applications. In European Symposium on Artificial Neural Networks (ESANN), 2016.

[7] Rudolf Szadkowski, Jan Drchal, and Jan Faigl. Basic Evaluation Scenarios for Incrementally Trained Classifiers. In International Conference on Artificial Neural Networks (ICANN), Lecture Notes in Computer Science, pages 507–517. Springer, 2019.

[8] A. Chilian and H. Hirschmüller. Stereo camera based navigation of mobile robots on rough terrain. In IEEE International Conference on Robotics and Automation (ICRA), pages 4571–4576, 2009.

[9] Dominik Belter, Jan Wietrzykowski, and Piotr Skrzypczyński. Employing Natural Terrain Semantics in Motion Planning for a Multi-Legged Robot. Journal of Intelligent & Robotic Systems, 93(3):723–743, 2019.

[10] Wei Mou and Alexander Kleiner. Online Learning Terrain Classification for Adaptive Velocity Control. In IEEE Safety Security and Rescue Robotics (SSRR), pages 1–7, 2010.

[11] Miloš Prágr and Jan Faigl. Terrain learning using time series of ground unit traversal cost. In 2019 Modelling and Simulation for Autonomous Systems (MESAS), pages 97–107, 2020.

[12] Miloš Prágr, Petr Čížek, Jan Bayer, and Jan Faigl. Online Incremental Learning of the Terrain Traversal Cost in Autonomous Exploration. In Robotics: Science and Systems (RSS), volume 15, 2019.

[13] Miloš Prágr and Jan Faigl. Benchmarking Incremental Regressors in Traversal Cost Assessment. In International Conference on Artificial Neural Networks (ICANN), Lecture Notes in Computer Science, pages 685–697, 2019.

[14] HyunRyong Jung, Arjun Menon, and Ronald C Arkin. A conceptual space architecture for widely heterogeneous robotic systems. In Biologically Inspired Cognitive Architectures (BICA), Frontiers in Artificial Intelligence and Applications, pages 158–167, 2011.

[15] Miloš Prágr, Petr Čížek, and Jan Faigl. Cost of Transport Estimation for Legged Robot Based on Terrain Features Inference from Aerial Scan. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 1745–1750, 2018.

[16] Chi-Yao Tseng and Ming-Syan Chen. Incremental SVM Model for Spam Detection on Dynamic Email Social Networks. In International Conference on Computational Science and Engineering (CSE), volume 4, pages 128–135, 2009.

[17] Y. Prudent and A. Ennaji. An incremental growing neural gas learns topologies. In International Joint Conference on Neural Networks (IJCNN), volume 2, pages 1211–1216, 2005.

[18] Jan Faigl and Petr Čížek. Adaptive locomotion control of hexapod walking robot for traversing rough terrains with position feedback only. Robotics and Autonomous Systems, 116:136–147, 2019.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.