Combining interaction and perception to determine the physical properties of the robot environment

Michał Bednarek

April 4, 2023

Contents

Abstract

The number of autonomous robot use cases significantly increased in multiple real-life scenarios, e.g., autonomous vehicles for transportation, safety and security, industrial inspection, or even space exploration. These are only a few examples from the plethora of new, inspiring, challenging domains where manipulators or mobile robots, e.g., walking machines, can be utilized. The real-life robotic use cases require robust perception, an intersection of sensor mechanics, hardware, and multi-modal data processing. Managing this complex system is often not trivial. Nevertheless, with autonomy comes responsibility and increasing demand for understanding the environment around an agent. The robot must adapt to unpredictable situations and cope with the open-world assumption. In this thesis, I advocate that the perception system robustness might be achieved by exploiting the haptic properties of the surroundings. There is a pressing need to explore this topic as the current solutions must cope better with this modality.

The following dissertation aims to create a unified robotic perception system that would meet the requirements of real-world applications and deliver appropriate information for onboard robot systems, e.g., localization, obstacle avoidance, or manipulation. Such a perception system has to be low-cost and computationally low-demanding due to the limited resources available on a mobile robot. A methodology concerns artificial intelligence, especially machine learning. These methods dominated the field of perception in recent years due to their superior performance, and their choice was natural. Deep learning methods focus predominantly on a supervised setup in material/terrain classification or when a continuous variable associated with a specific parameter is needed (e.g., stiffness estimation). Moreover, I also explored an unsupervised learning setup as it became a fundamental tool for understanding hidden characteristics in sensory data. Eventually, perception robustness emerged as a crucial characteristic of deploying it in the real world. The dissertation includes multiple solutions to increase the robustness, primarily by using attention modules.

Streszczenie

Percepcja robotyczna jest intensywnie rozwijającą się dziedziną, leżącą na styku sprzętu, mechaniki i wielomodalnego przetwarzania danych, które często nie są łatwe do wykorzystania. Można zaobserwować, że liczba autonomicznych robotów znacznie wzrosła w wielu dziedzinach życia, takich jak autonomiczne pojazdy transportowe, bezpieczeństwo, inspekcja przemysłowa czy nawet eksploracja kosmosu. To tylko kilka przykładów z mnóstwa nowych, inspirujących, ale wymagających dziedzin, w których roboty będą wykorzystywane do manipulacji i przemieszczania się. Niemniej z autonomią wiąże się pewna odpowiedzialność i rosnące zapotrzebowanie na zrozumienie środowiska otaczającego robotycznego agenta, tak aby mógł się dostosować do nieprzewidywalnych sytuacji. Obecnie systemy percepcji jedynie w niewielkim stopniu rozpoznają właściwości taktylne otoczenia.

Celem rozprawy jest stworzenie systemu percepcji robotycznej, który spełniałby wymagania rzeczywistych zastosowań i dostarczał informacje dla systemów działających na pokładzie, takich jak lokalizacja, unikanie przeszkód czy strategia manipulacji. Ponadto, system musi być tani i mało wymagający obliczeniowo ze względu na ograniczone zasoby robota mobilnego. Metodyka pracy dotyczy przede wszystkim sztucznej inteligencji, a zwłaszcza uczenia maszynowego. Metody te zdominowały tę dziedzinę ze względu na najlepsze wyniki, więc ich wybór był naturalny. W niniejszej rozprawie przedstawione zostały metody głębokiego uczenia nadzorowanego w dwóch typach zadań. Po pierwsze w klasyfikacji powierzchni dotykanej przez robota. Po drugie w zadaniu regresji parametrów fizycznych tejże powierzchni, do których zalicza się np. sztywność dotykanego obiektu. Uczenie nienadzorowane natomiast jest kluczowym narzędziem do zrozumienia ukrytych właściwości danych sensorycznych oraz tego, jak robot może je wykorzystać bez nadzoru. Także odporność percepcji robota okazała się ważną cechą dla wdrożenia systemu w rzeczywistości. Poniższa rozprawa zawiera rozwiązania mające na celu jej zwiększenie, głównie poprzez wykorzystanie modułów atencji.

Acknowledgments

I want to take this opportunity to express my sincere gratitude to many people who have been instrumental in my journey toward completing this Ph.D. thesis.

First and foremost, I would like to thank my auxiliary Ph.D. supervisor, Krzysztof Walas, for his long-term cooperation in research projects, scientific work, opportunities to visit the best places to grow as a researcher, and his unwavering support during the tough times. I would also like to express my sincere appreciation to my other Ph.D. supervisor, Paweł Drapikowski, for his guidance throughout the process of writing this thesis, his invaluable reviews, and his kind words of encouragement.

I want to acknowledge the tremendous contributions to the work of my co-authors of my research papers and other colleagues at the Institute of Robotics and Machine Intelligence, who have provided me with a stimulating research environment and countless opportunities to collaborate and learn. With fruitful discussions, brainstorming, and endless conversations with you on various topics, that work would be complete with the conclusions drawn during them.

Lastly, I am deeply grateful to my family, especially my wife, who has always been by my side. Thank you for your unwavering support, understanding, and encouragement throughout this challenging journey. Your love and support have been a constant source of strength and inspiration for me.

Abbreviations

AGV	Atomated Guided Vehicle.
AHRS	Attitude and Heading Reference System.
AUC	Area Under Curve.
Bi-LSTM	Bidirectional Long-Short Term Memory.
BiGS	BioTac Grasp Stability Dataset.
BiGS V2	BioTac Grasp Stability Dataset V2.
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies.
BOSS	Bag-of-SFA Smbols.
CNN	Convolutional Neural Network.
CoG	Center of Gravity.
COTE	Collective of Transformation-Based Ensembles.
CPU	Central Processing Unit.
DEC	Deep Embeddings Clustering.
DEI-DEO	Decision In-Decision Out.
DTW	Dynamic Time Warping.
EDA	Exploratory Data Analysis.
EE	Elastic Ensemble.
EM	Expectation-Maximization.
F/T	Force / Torque.
FC	Fully Connected.
FFT	Fast Fourier Transform.
FIFO	Feature In-Feature Out.

GMM	Gaussian Mixture Model.
GPU	Graphics Processing Unit.
GRF	Ground Reaction Force.
HAPTR	Haptic Transformer.
HaTT	Penn Haptic Texture Toolkit.
HIVE	Hierarchical Voting.
IMU	Inertial Measurement Unit.
KL	Kullback - Leibler.
KNN	K Nearest Neighbour.
LiDAR	Light Detection and Ranging.
LMF	Low-Rank Multimodal Fusion.
LSTM	Long-Short Term Memory.
MAE	Mean Absolute Error.
MAL	Modality Attention Layer.
MAPE	Mean Absolute Percentage Error.
Mid	Intermediate Fusion.
MLP	Multi-Layer Perceptron.
MoE	Mixture of Experts.
PHAC-2	Penn Haptic Adjective Corpus 2.
PUTAny	Poznan University of Technology ANYmal dataset.
PVC	Polivinyl Chloride.
QCAT	Queensland Civil and Administrative Tribunal dataset.
RGB	Red Green Blue.
RGB-D	Red Green Blue - Depth.
RNN	Recurrent Neural Network.
ROC	Receiver Operating Characteristic.
ROCKET	Random Convolutional Kernel Transform.

SLAM	Simultaneous Localization and Mapping.
SVD	Singular Value Decomposition.
SVM	Support Vector Machine.
TCN	Temporal Convolutional Network.
TSC	Time Series Classification.

Content

Chapters 2 and 3 present tactile and kinesthetic sensing solutions for robotic arms and walking robots. In both robotic setups, an agent interacted with a nondeterministic environment, such as deformable objects or rough terrains with different material classes. All developed perception systems are based on deep learning methods. When writing this thesis, they achieved state-of-the-art classification, recognition, and representation of learning results. Methods for material classification and object stiffness regression using a soft gripper [1] were taken on in Chapter 2. Moreover, this Chapter also investigates the unsupervised learning approach for haptic data grouping without supervision. This task is an emerging topic in the robotics community and a way to tackle learning from the tremendous amount of available datasets without the manual labelling process, which is time-consuming, expensive, and tedious. Chapter 3 presents the research conducted in the walking robots area, including the classification of terrains using F/T sensors feedback or orientation provided by the IMU. All experiments in this Chapter were performed with a high emphasis on generalization for different robots and rapid inference time. That Chapter also presents the Modality Attention Layer (MAL) as a solution for the weighting input modalities concerning their *significance* level, which results in improvements in a system's robustness against input data degradation. Chapter 3 presents the multi-modal fusion methods using deep learning. Multiple state-of-the-art predictive models were tested on three, separate datasets prepared for different robotic, perception-related tasks – manipulation of deformable objects [2], texture recognition [3], and multi-label classification of haptic adjectives [4]. These datasets include time-series data from different (often non-homogeneous) modalities, e.g., video streams, and Force / Torque (F/T) sensors, where it is not a trivial task to couple them together. The work tackled the problem of multi-modal fusion by measuring the overall performance of different deep neural networks in the selected tasks. Moreover, the research includes examining the impact of sensor failures and noise in the input data on the final results

– looking at the problem from the robotics perspective. The results from this work gave a highly valued insight into the fusion of different modalities in the data-driven approaches to robotic tasks.

Chapters in Part III contain experimental verification of developed/adapted methods. Chapter 5 shows the comparison of different approaches to the object's stiffness estimation using a simulated dataset and real-world data samples. It focuses on the Recurrent Neural Network (RNN) and the number of real-world data samples included in the simulation dataset to fill in the *reality gap* – a common phenomenon present in data-driven approaches trained in the simulated environments. Moreover, the experiments towards unsupervised learning utilized the Deep Embeddings Clustering (DEC) method [5]. Chapter 6 presents the Haptic Transformer (HAPTR) - a novel and lightweight approach that utilizes attention modules to classify the terrain samples based on haptic/inertial time series. In the following work, the HAPTR was compared to state-of-the-art algorithms and achieved high performance in this task. Chapter 7 presents an extensive comparison of multi-modal fusion methods – Late Fusion, Mixture of Experts (MoE), Intermediate Fusion (Mid), and Low-Rank Multimodal Fusion (LMF), with an emphasis on the significance of the feature level fusion and data degradation robustness.

In Chapter IV final remarks were given, a summary with discussion on results and plans for future research. All chapters in the following dissertation include a domain-specific literature review.

Part I

Introduction

Chapter 1

Motivation

1.1 Broad perspective

In the coming years, we will observe a rapidly growing market for ground robots that can operate in many different environments. The idea that mobile robots could assist humans in various tasks started penetrating people's awareness. Many factors influence the development of advanced robotics systems. Firstly, we have a global viewpoint based on the competition between countries, research facilities, or commercial companies and the market's response to that phenomenon. Secondly, the scientific perspective – seen as the progress in robotics perception research, inspires more and more people to be engaged in that competition.

Nowadays, large companies compete in the high-tech industry, accelerating the development of robotic innovations. Anybotics's ANYmal [6] or Boston Dynamics' Spot are great examples of mobile robots' maturity, and we can expect that they will soon be deployed in significant numbers. From a global standpoint, innovations create new markets and supply chains. That, in turn, improves the strategic position of a facility that controls a new technology. Let the development of the space industry be an example. For many years wheeled robots (e.g., lunar rovers) served different space agencies in their missions to gather new information about planets in the Solar System. However, there is still no autonomous system capable of exploring, e.g., steeper slopes – crucial places from the geological point of view. Wheeled systems are unprepared to adapt well enough to rough, inclined and unpredictable terrain. However, the answer to that problem would be a legged platform such as [7] that interacts with its environment and could adapt its gait to the unpredictable ground surface.

Additionally, such a robot should also autonomously recover from any fall. Its perception system should properly recognize terrain properties using all the available sensory information. Even if there is a limited demand for this type of system right now, with great confidence, it will appear in the coming years. One can observe the increasing competition between global, high-tech players in space exploration. The company that will provide such a solution will make a tangible profit in the future, creating commercial potential for developing robotics perception systems.

On the other hand, the rapid development of robotic perception systems can also be more down-to-earth. Firstly, sensors have become more affordable in recent years, which has engaged more and more research groups in the field. It is a scientific breakthrough because of the popularity of advanced sensory systems, e.g., F/T measurement devices, Light Detection and Ranging (LiDAR) scanners, 2D laser scanners, depth cameras, or a variety of inertial sensors such as Inertial Measurement Unit (IMU) or more sophisticated Attitude and Heading Reference System (AHRS). All of them pushed forward the pace at which new technologies in robotics appeared over the last few years. Innovations in sensory systems and the increasing computational power of modern processing units brought an opportunity to apply existing algorithms in the real world, find their limitations, and improve or replace them with new ones.

A robust robotic perception became an emerging field at the level of governments and corporations creating the global market. Additionally, the accessibility to the knowledge and required hardware started to be relatively easy. Taking both factors into account, the author of the following dissertation believes that now is the right time to pursue research in that topic – *it is not too early, nor too late* [1].

Information about a robot's environment is crucial when developing a robust system that would let a robot autonomously adapt to the current circumstances or explore undiscovered knowledge. Nowadays, robots utilize geometric information about its surrounding, and they use camera images, depth maps, or point clouds – namely visual sources. Multiple research areas use these sources of visual information, like Simultaneous Localization and Mapping (SLAM), path planning, obstacle avoidance, scene reconstruction, or manipulation of elastic objects. Until the robotic agent's surroundings are static, rigid, and predictive, the perception problem might be solvable without any high-level understanding. Nowadays, many semi-autonomous machines operating in highly controlled environments use only geometric information about a scene. A great example

[1] Panel *How to be a Good Citizen of the CVPR Community* – a talk by Vladen Koltun.

of such applications is an Atomated Guided Vehicle (AGV) used chiefly in logistics and can successfully navigate in 2D maps using laser scanners. However, such robots are prone to errors whenever their environment stops being static, e.g., when the human enters the zone or a hall's arrangement changes. Taking such cases into account, recently, more and more researchers have noticed that the environment's geometry is not enough to achieve a robust perception, especially when considering the deployment of a robot in the human world. A high-level understanding also matters and is vital to the perception system. Understanding the semantics of the scene is crucial to interact with it because it gives the cues if the robot can complete the task and, if so – how to behave or interact with the environment to complete a mission.

1.2 Problem statement

The robotic agent must have some internal representation of the world around it to adapt to and overcome environmental conditions – the notion of physical properties as a mass, friction, stiffness, or more categorical, e.g., a class of an object or a surface. However, it is challenging to provide such representation, as mobile robots operate in unstructured and often unpredictable environments, and their perception systems must handle many disturbances. Data-driven methods are the most popular yet most effective approaches for representation learning. Typically, they operate on learnable features where the recognition task is more facilitated than using raw sensory feedback. Nevertheless, they suffer from multiple drawbacks. Firstly, they require a vast amount of data to generalize. For example, the terrain recognition system trained on the dataset with samples gathered only on the flat surface probably will not be fully operational when slopes appear. Secondly, the noise in the training data might mislead the perception system and turn the training process into overfitting or being sensitive to sample-specific glitches. The thesis's main objective is to explore and implement a system composed of functional blocks that would enable and further improve the overall quality of robotic haptic sensing. The goals of this dissertation allowed for the formulation of scientific theses, which gave a present shape to this work:

Haptic robotic perception, defined as a capability of a robotic agent to recognize complex patterns in tactile and kinesthetic sensations, can be achieved using deep learning methods because they exhibit superior performance in a vast majority of automated tasks. That ability can be further extended by unsupervised learning meth-

ods that let analyze the incoming data and measure its similarity to previously seen examples, which improves a high-level understanding of the robot's environment.

The supporting theses follow the statement of this dissertation that refer to particular gains brought to the robust perception system as a result of this research:

- Typically, haptic signals are time-series, e.g., forces, torques, or inertial measurements representing the motion of a sensor in space. Posing the problem of haptic perception as a sequence-to-sequence task would enable us to use an attention mechanism that produces weights for each sample in the input signal. Feeding such weighted signals to a supervised learning method would simplify the task of, e.g., terrain classification and improve the robustness of a proposed system. Such a classifier could focus more on relevant parts of its input.

- An understanding of the robot's environment will improve based on the information from multiple modalities, e.g., F/T signals, images, or raw sensory signals from haptic electrodes. We can fuse data from heterogeneous modalities using deep learning methods, which do not restrict the proposed system to homogeneous data only.

> Chapters of the following dissertation present the author's research contributions, most of which have already appeared in peer-reviewed scientific journals and conferences from 2019 to 2022. Every Chapter includes a heading with listed publications that served as a base for the presented work. The copyright sign with the year of publishing, e.g., (© 2019 - 2022 ELSEVIER), stands next to published graphics and tables if the publisher's policy requires that for reusing for non-commercial purposes such as including in a dissertation.

Part II

Methods

Chapter 2

Material recognition

> The following Chapter and the corresponding results in Part III are partially based on two publications of the author of this thesis.
>
> - (© 2019 IEEE) J. Bednarek, M. Bednarek, P. Kicki, K. Walas (2019) "Robotic Touch: Classification of Materials for Manipulation and Walking". In: IEEE International Conference on Soft Robotics (RoboSoft)
> - M. Bednarek, P. Kicki, J. Bednarek, K. Walas (2021) "Gaining a Sense of Touch Object Stiffness Estimation Using a Soft Gripper and Neural Networks" in Electronics Journal

2.1 Introduction

Human perception of the world highly depends on its physics. Our sensory system provides haptic sensations to the brain, which lets us, e.g., plan a direct trajectory to grab a cup of tea, squeeze a wet sponge or flip a book page. We know how to do these things and predict objects' deformations based on their physical properties. Moreover, our hands constitute highly effective grippers, outperforming industrial ones in dexterity. Taking into account our assumptions about the world that come from our minds, combined with the *embodied intelligence* [8] of our hands, we can perfectly adjust the manipulation process to varying external conditions. However, machines do not have such in-built proficiency, and their abilities to manipulate allow only for managing repetitive tasks.

Soft robotics is a sub-field of robotics that focuses on robot design, its locomotion, and real-world interaction with the environment utilizing a paradigm known as the *intelligence by mechanics* [8]. It assumes that the robot's body should adapt to the surroundings through decentralized mechanical system-environment interactions rather than one decision-making unit, e.g., an artificial neural network. The aftermath of the research on this topic is biologically inspired soft grippers [9, 1, 10, 11], designed from soft materials in such a way that can adjust their shape to the grasped objects, which allows them manipulating delicate and deformable objects. One can state that the shape of that gripper corresponds to the physical properties (e.g., stiffness, elasticity, roughness) of grasped objects. However, these soft grippers are sensorless by design, and measuring their deformations was not the researcher's area of interest.

On the one hand, one can observe the growing number of available applications of sensors capable of capturing high-dimensional deformations of soft and unpredictable physical objects [12, 13, 14]. However, they do not directly measure the behavior of the soft gripper, but only the properties of the surroundings or grasped objects. In the following dissertation, I argue that we can use traditional and widespread sensors based on microelectromechanical systems to predict the physical nature of the robot's surroundings by estimating the gripper deformation. Thereby, the following chapter presents findings about a supervised, hybrid approach that connects an *embodied intelligence* of a soft gripper with an *artificial intelligence* system to provide an easy-to-use, open-source and inexpensive method of estimating the physical properties of objects with various stiffness parameters.

The sensors used in material classification are primarily accelerometers [15], pressure mapping sensors [16], 3D force [17] and 6D F/T sensors [18]. Most approaches presented in the literature use very controlled environments when performing experiments, as in the case of accelerometers [15]. Currently, a preferred approach to material recognition is by a programmed set of exploratory movements instead of a single touch [17, 16]. However, the desired behavior is to achieve a high material classification score in an uncontrolled environment during a normal robot operation without performing particular movements. The classification should be on the raw signals without additional pre-processing and hand-crafted representation.

As machine learning methods became more accessible to most researchers, these learning-based methods constituted a typical way to improve advancements in particular domains. However, some might argue that embodied intelligence will follow that trend, as it was usually outside the scope of mainstream research topics [19]. Robotics is a unique field that almost always requires some form of interaction with the agent's environment, which might prevent unlimited usage of data-driven methods, which happened in other areas related to, e.g., computer vision or natural language processing. While the autonomous robot should also operate in unknown environments, another limitation of learning-based algorithms is their need for adaptation to previously unseen scenarios. This restriction leads to the limited capability of real-world deployment except for a narrow set of fields. Yet, there is no simple answer to these problems because no autonomous machine can be prepared for every event. The author of the following thesis argues that we should not try to be ready but actively adapt to changing conditions. In such a setup, unsupervised learning might improve that high-level understanding of the robot environment, where the haptic recognition of materials is a good starting point. There needs to be more research on robot environment understanding using unsupervised learning methods.

The following chapter investigates multiple approaches to the material and object recognition task. The stiffness estimation problem was tackled using various deep learning methods, including convolutional and recurrent architectures. Moreover, the chapter presents novel haptic datasets for the stiffness estimation based on real-world and simulated IMU signals from sensors attached to the fingers of an elastic gripper. Secondly, the author of the following thesis delved into the challenging problem of unsupervised learning for the haptic recognition task, where the clustering model did not have any prior knowledge about input signals. Typically, these algorithms analyze unlabeled datasets to group them and discover hidden patterns. In robotics, such systems would substantially improve data association of previously seen places, objects, or finding anomalies. However, scientific publications in that field with real-world robotic setups are still limited, even though unsupervised learning is an emerging field of Artificial Intelligence.

2.2 Related work

2.2.1 Stiffness measurement

In [20], authors proposed a practical application for a continuous rail rigidity measurement using the accelerometer and oscillating mass on the rolling wheel. This work indicates that the issue under examination is vital in engineering. Unlike the previous method, the authors of [21] propose to use the non-contact measurement of spindle stiffness. The authors suggested a magnetic loading device that enabled them to make measurements while the spindle rotated. Due to the usage of magnetic loading, that method limits itself to ferromagnetic objects. Measuring a stiffness was also possible at a much smaller scale than presented in [22]. The authors reviewed the nanoindentation continuous stiffness measurement techniques and applications. The range of stiffness coefficients of materials is extensive. To avoid saturation and enhance precision, the authors of [23] proposed a portable measurement device able to adjust a sensing range by manipulating tool parameters, such as touch module separation, indenter protrusion, and spring constant of the force sensing module. Authors of [24, 25] analyzed the stiffness measurement techniques applied to the polymer foams, which are similar to those used in this paper. In [24] there, was proposed a procedure for measuring stiffness using dot markers on object surfaces and compression plates to exert a force on an object. Authors stressed that non-axial compression tests resulted in worse performance, which was usually the case in robotic manipulation. Similar to the proposed solution, in [26], authors proposed the IMU-based approach. However, in a different task – the reconstruction of the configuration of a soft gripper. As opposed to that work, the experimental section of this thesis describes the proposed indirect measurements of the object's stiffness property by the change of behavior of the soft gripper while squeezing, not the gripper's configuration itself.

2.2.2 Stiffness estimation

The work [27] presents an alternative approach that does not require measuring the object deformation. The authors proposed the Candidate Observer-Based Algorithm, which exploits two force observers, with different stiffness candidates, for estimating the stiffness of objects with complex geometry. Unfortunately, the authors did not refer to the ground truth stiffness measurements. However, such a comparison was made in [28]. The neural network was trained to predict the stiffness coefficient based on a maximum penetration and maximum contact pressure variation. Authors of [4] presented an alternative deep learning strategy for understanding the haptic properties of objects. The real-world objects were classified in the set of haptic adjectives in the multi-label fashion based on haptic signals from BioTac sensors [29] and images. That work showed a correlation between haptic sensor readings and the structure of real-world objects, and this fact was utilized in the following work. The extensive overview of machine learning methods in the soft robotics aspect was described in [30]. The authors distinguished between sensor characterization and systems characterization. In the group of sensor characterization, the use of RNNs for parameters regression is widespread, as we are dealing with signals and continuous values of sensor parameters. On the system characterization level, one is more focused on high-level labels, successful grasp [31], or slip detection. However, the classification of signals with categorical values is more common. In [32] learned, control mechanisms were used, reinforcement learning [33] or learned differentiable models [34].

2.2.3 Processing inertial data

The popularity of IMU stems from its availability and low price. One possible use in the robotics community is a robot's state estimation. In [35], acceleration and angular velocities collected from sensors located on the humanoid leg and joints' angles were used to estimate the velocities of robot links. Authors in [36] presented multiple interesting approaches for indirectly measuring ground reaction forces during a human walk using wearable IMUs. The other field that often utilizes acceleration is material classification. In [3], authors used the haptic device SensAble Phantom Omni [37] to gather accelerations and velocities while scratching material surfaces. The authors of [38] used that dataset to train a deep Convolutional Neural Network (CNN) to map from raw signals to textures' classes. The presented method stays close to our solutions for stiffness estimation.

2.2.4 Fabric recognition

The authors of [39] proposed a material classification solution that basis on raw acceleration gathered during exploratory moves of a sensor mounted at the tip of the rigid tool. Similar to our material classification method, data gathering was carried out by authors of [17] based on readings from the optical force sensor. The field of tactile material recognition also draws inspiration from biological systems, which was described in [16]. The authors used a skin-like flexible pressure-sensitive sensor. This method exceeded human-level performance in material recognition and differentiation, based only on pressure distribution signal processing. All of the works mentioned above were deep-learning-based and trained in a supervised manner. In [40], a different approach was proposed – the authors made an architecture based on Generative Adversarial Network [41] in a semi-supervised fashion, which enabled a robot to learn from unlabeled data or even adapt itself to the previously unseen types of material. Sensing a force when the model predictive control framework gathers haptic data samples becomes crucial. Authors of [42] proposed a robotic dressing assistant for people with disabilities, where a RNN predicts force signals to be applied while performing this type of task.

2.2.5 Unsupervised robotic perception

Unsupervised learning is currently the most emerging yet least explored field of artificial intelligence. However, its practical applicability remains limited, especially in such a demanding problem as tactile perception. It is due to challenging optimization and interpretation of obtained results, especially in the clustering assignment, where the model needs to provide explicit information about the grouping criteria. The specific domain of application of unsupervised learning is medical assignments, like emotion recognition using electroencephalography [43] or finding patterns in medical data analysis to predict seizures [44]. Gesture recognition is a related category where these methods found an application. In [45], the authors proposed a trajectory segmentation method for surgical applications that detects critical trajectory points that define relevant segments. In [46], a robotic assistant recognizes the gestures of a surgeon.

In robotics, place recognition is a typical application of unsupervised learning [47, 48, 49, 50]. In [51], the authors tackled a related problem of in-sequence condition changes using CNN-based descriptors and unsupervised learning methods. In this work [52], authors reviewed numerous research articles to determine the number of so-called perceptual dimensions, i.e., the number of dimensions in which humans can perceive the structure. They concluded that there are five fundamental dimensions of tactile perception – macro and fine roughness, warmness/coldness, hardness/softness, and friction (moistness/dryness, stickiness/slipperiness). In the following dissertation, these categories describe clusters created by the unsupervised learning method in the tactile recognition assignment. Commonly, unsupervised learning methods serve as feature extractors, especially in the haptic adjectives classification that describes some physical properties of objects. In [53], the authors presented their research on tactile understanding and haptic perception using unsupervised feature learning methods. In this follow-up work [54], they proposed a method for predicting the so-called perceptual distribution of a haptic adjective based on dictionaries of these features. The term perceptual distribution relates to the human-like understanding of tactile sensations perceived gradually and not restricted to any predefined set like in, e.g., binary classification. The work [55] proposes a similar approach to the one presented in the following Chapter, where the authors proposed an interactive method for classifying previously unseen objects. In their experiments, a robot grasps an object and categorizes it as novel or unseen using a so-called One-Class Support Vector Machine (SVM). Then, it creates local tactile representations for new instances and learns a new unsupervised model. After that, the process starts over with another object. However, this method requires guided exploratory robot grasping based on the custom-made pressure sensor. At the same time, the approach proposed in this Chapter relies purely on non-guided touching episodes and corresponding force measurements gathered using the low-cost sensor.

2.3 Proposed solution

Most state-of-the-art recognition systems basis on artificial neural networks and need substantial data volume for appropriate training in a given task. However, creating a dataset could comprise a troublesome problem in robotics, as it requires a real-world robot to be involved and manual labeling in the case of supervised learning, which is a tedious and labor-intensive process. Typically such experiments appear to be time-consuming and hard to design due to the limitations of a robotic platform. In the wide range of recognition tasks, one approach to such data is to build a predictive model that works in an unsupervised or semi-supervised manner. Typically, such approaches are part of an explainable artificial intelligence domain and primarily focus on a data representation analysis [56]. However, a lack of interpretability in robotics might be a troublesome issue. However, to the author's knowledge, such solutions still need to be explored in the robotics community. The problem of supervised haptic recognition might be solved using numerous robotics simulation toolkits. Generating a large amount of data in the simulation and mixing it with a small portion of the real-world samples would be a recipe for a valid dataset for the neural network that needs to operate in the real world. However, such a case must urgently cover the reality gap. No currently available simulation environment would fully resemble the real world.

This Chapter investigates multiple data-driven approaches for a material recognition task from raw sensory feedback, including force sensors and inertial units. It consists of two thematically divided sections, each with a different approach to the recognition task. Firstly, it presents the supervised learning model for stiffness regression, which uses simulated and real-world data samples. The main focus was on different types of RNNs with feed-forward modules (convolutional or Fully Connected (FC)) that served mainly as input feature encoders or final layers. The Chapter investigates the problem of the so-called reality gap in the robotic perception system. The simulated environment served as a data generator for data-hungry recognition methods.

Additionally, the Chapter presents results obtained from the experiments on Exploratory Data Analysis (EDA) on haptic signals from different robotics assignments using unsupervised learning methods. Typically, that kind of data analysis does not have any clear goal and aims to find the relationship between data samples, discover clusters, and any new insights that will help to improve data understanding. In machine learning, one of the core concepts in the EDA is the clustering assignment that enables finding groupings in data. Experiments in the following Chapter focused on finding the best clustering algorithm for haptic signals obtained from touching different materials or objects. All datasets involved in investigations were primarily designed for supervised tasks, containing actual labels for each sample. The central presumption was that the expected number of classes discovered in the data was the number of authentic classes. Due to this, one could compare different clustering strategies based on metrics that require true labels: clustering accuracy, normalized mutual information, and purity.

2.4 Supervised stiffness estimation

2.4.1 Problem formulation

Let $f\colon \mathcal{M}_{IMU} \mapsto \mathcal{K}$ be a stiffness estimation function that maps elements from the domain of IMU measurements $\mathcal{M}_{IMU}$ to the positive real-valued counter-domain of the material stiffness $\mathcal{K}$. IMU measurement $M \in \mathcal{M}_{IMU} = \mathbb{R}^{2\times 6}$ is defined by vectors of linear accelerations $\mathbf{a_n}$ and angular velocities $\boldsymbol{\omega}_n$ organized row-wise in the matrix M, where the n-th row is $[a_{x_n}, a_{y_n}, a_{z_n}\omega_{x_n}, \omega_{y_n}, \omega_{z_n}]$, and $n = \{1, 2\}$ because the setup consists of two IMUs attached to two fingers of the gripper used in the experimental section. The set $\mathcal{K}\colon \{k \in \mathbb{R}^{+}\}$ is defined as scalar stiffness parameters of squeezed objects expressed in $\frac{N}{m}$. Given the definitions above, the following elements of the work are:

- three estimation methods $f'_{conv}, f'_{lstm}, f'_{bilstm}$, where each $f'\colon \mathcal{M}_{IMU} \mapsto \mathcal{K}$ approximates the function f using a different deep neural network architecture;

- two datasets $\mathcal{D}_{real}, \mathcal{D}_{sim}$ that consist of real-world and simulated data samples of pairs $d\colon (M, k)$, where $M \in \mathcal{M}_{IMU}$ and $k \in \mathcal{K}$;

2.4.2 Real-world dataset

The Yale OpenHand shown in Fig. 2.1 is the under-actuated, two-finger soft gripper with joints in the form of urethane elements to assure the elasticity of fingers. The real-world model was 3D printed and driven by hobby servos capable of generating a force up to 10N. Fingertips of the hand had mounted IMUs, which measurements m served to estimate grasped objects' stiffness k. The following Chapter investigates how the *embodied intelligence* of such a soft gripper could be utilized alongside the *artificial intelligence* system to predict a squeezed object's real-world stiffness coefficient k. The following paragraph presents the process of creating the dataset $\mathcal{D}_{real}$.

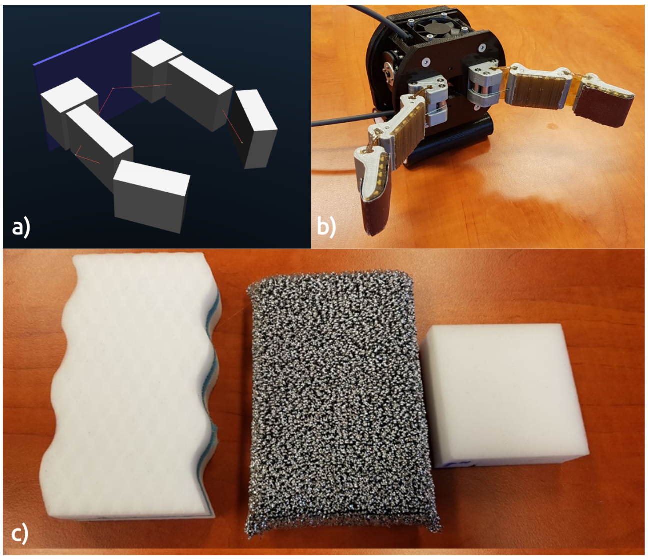


Figure 2.1: The real-world scenario involved the 2-finger Yale OpenHand gripper [9]. A sufficient number of training samples for the learning process had to be ensured. This gripper model was prepared in the MuJoCo simulator as depicted in *a)*. In *b)*, real fingers consist of three plastic blocks with flexible parts made of urethane. In *c)*, there are presented examples of sponges, exposing different stiffness, used in our real-world experiments.

Table 2.1: Stiffness coefficients were computed for five different objects using the presented procedure.

Object	Stiffness [N/m]
Wire sponge	909
Hard sponge	1020
Polish sponge	735
Soft sponge	380
Squash ball	1353

First, a coefficient k was estimated for real-world objects in the $\mathcal{D}_{real}$. The robot had a 3D-printed plastic bar mounted at the flange and pressed objects with the desired force using the Dynamic Force Control mode to obtain ground-truth values. In that mode, the manipulator accurately measured the displacement under specific forces. Thus, k was computed according to the Eq. 2.1, where f_1 and f_2 are forces in Z-axis while pressing an object with a tool and $|d_1 - d_2|$ is the relative distance that corresponds to the deformations under f_1 and f_2 respectively. The chosen objects express nonlinear behavior in their stiffness model (e.g., the greater robot compresses the sponge, the less deformation it adds). However, objects did not reflect nonlinear effects in the specified range of exerted forces. Therefore, the estimated stiffness is considered homogeneous for the entire object. Tab. 2.1 contains different stiffness coefficients $k \in \mathcal{K}$ measured experimentally for each object.

$$k = \frac{|f_1 - f_2|}{|d_1 - d_2|} \in \mathcal{K} \tag{2.1}$$

Afterward, the Yale OpenHand was used to perform the squeezing motion of each object and 500 IMU signals $m \in \mathcal{M}_{IMU}$ registered during motions were collected. Each signal m in $\mathcal{D}_{real}$ consists of 12 sensor readings. Each of them was 200 time-steps long. In $\mathcal{D}_{real}$, each object is equally represented by 100 samples and split into two subsets – 200 train and 300 test samples for the sim-to-real experiments. Both subsets in all the experiments remain unchanged. The following claim was made to address the physical interpretation of obtained parameters k taking as input measurements m: The motion of gripper fingers registered while squeezing different objects would significantly vary, as shown in Fig. 2.2. One can observe that depending on the object's stiffness, the magnitude, and oscillations of both - angular velocity $\boldsymbol{\omega}$ and linear accelerations $\mathbf{a}$ were significantly different from each other, i.e., in the range of values or an oscillation rate. Taking that phenomenon into account, this Chapter puts forward the thesis that it is possible to distinguish between different stiffness parameters in the space of IMU signals registered during the squeezing of these objects.

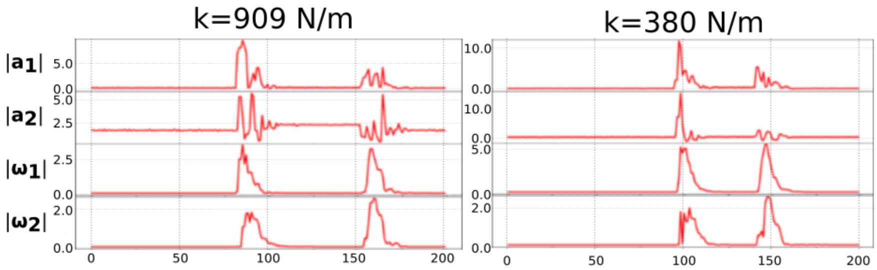


Figure 2.2: Comparison between exemplary samples from the real-world dataset while squeezing objects with different stiffness values with a soft gripper. Values $|\boldsymbol{a}_1|$, $|\boldsymbol{a}_2|$, $|\boldsymbol{\omega}_1|$ and $\boldsymbol{\omega}_2|$ refer to magnitudes of accelerations and angular velocities registered by two IMUs and are expressed in $\frac{m}{s^2}$ and $\frac{rad}{s}$ respectively.

The number of samples, as well as distinct objects, remained limited. The root cause was the soft gripper's time-consuming data-gathering process and hardware limitations. In that situation, the number of different labels was insufficient to perform a successful regression without overfitting. In fact, with such a low diversity of accessible objects, a regression task would inevitably turn into a classification undesired in the stiffness estimation. A second dataset was based on the simulation to overcome that problem, where more training samples could be generated. Stiffness coefficients were adjusted to meet measured values.

2.4.3 Simulated dataset

Data-driven models frequently suffer from the limited ability to generalize to new domains outside their training dataset. However, the rising popularity of deep learning algorithms in the robotics community led to a significantly increased need for data. The state-of-the-art approach is to perform experiments in simulation and use the gathered data to feed neural networks. In a wide range of robotics applications, researchers can choose from many available physics simulators. In our case, the MuJoCo physics simulator was selected due to its efficient implementation regarding soft object modeling, which is troublesome in other simulators, e.g., in the PyBullet [57] or Gazebo [58]. Fig. 2.3 presents the simulated soft-robotic gripper. Tendons connect fingers and are pulled by the actuator, which simulates the pneumatic cylinder. The simulation model basis on the 3-finger real gripper [1], but with one finger removed. As it is depicted in Fig. 2.3a, the $\mathcal{D}_{sim}$ consists of samples when the gripper squeezed and released objects of three shapes - a ball, a box, and a cylinder, all with a variable k. For the elastic deformations simulation of the gripper, each geometrical block of each finger was connected to others by three hinges. In this setup, ranges of each joint in a roll, pitch, and yaw axes could be adjustable, as was depicted in Fig. 2.3b. Finally, each behaves similarly to the elastic finger.

For clarity, a stiffness parameter k, in $\mathcal{D}_{sim}$ was defined the same way as the MuJoCo simulator – as the stiffness of a spring attached to a CoG of a geometrical block and its surface. The experimental section always assumes that the object is homogeneous.

The collection of data proceeded accordingly. Firstly, the object appears between fingers, and the actuators start to close the gripper and squeeze it. Then, in half a duration of a squeeze episode, the gripper opened up. While this process, an object was embraced by fingers that adapted to its shape. A stiffness coefficient k was expressed in $\frac{N}{m}$ and varied among episodes to equally cover the range (from 300 to $1400\frac{N}{m}$), which fits the real-world data range. Masses of gripper parts were adapted to meet real-world values. Similar to ob-

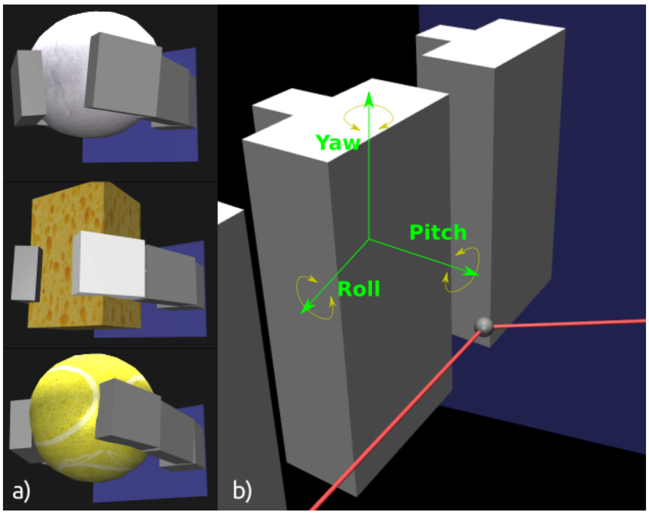


Figure 2.3: Soft-robotic gripper in the MuJoCo environment: *a)* the gripper squeezes and releases objects in three shapes - a ball, a box, and a cylinder, all with a variable stiffness parameter; *b)* each geometrical block of each finger is connected to others by three hinges. In this setup, one can easily adjust the ranges of each joint in roll, pitch, and yaw axes.

jects' mechanical impedance, their dampings, stiffness of all joints and springs in a system. Two IMUs were placed on a MuJoCo's element called *site* and located in the 3/4 of the length of each finger on the outside surface. The first from two simulation datasets resembled the real-world data and consisted of 5000 training-validation samples from squeezing the box object. Its purpose was to enrich real-world data. The second simulation dataset consisted of three shapes: boxes, cylinders, and spheres. It counted 3999 training-validation samples, which gave 1333 samples per object. It served to verify whether any of the neural networks f' can avoid overfitting to any particular shape. Additionally, the model's generalization abilities were investigated using three test datasets – 133 samples for each object.

2.4.4 Recurrent neural network architecture

In Section 5.1 on the experimental evaluation, the main task of proposed neural networks was to approximate the stiffness estimation function f from fixed-length sequences $m \in \mathcal{M}_{IMU}$ of accelerations and angular velocities measured by IMUs. This research proposes to test three types of neural networks – the CNN f'_{conv} based entirely on 1D convolutional blocks, the CNN-LSTM f'_{lstm} with forwarding Long-Short Term Memory (LSTM) units [59], and the CNN-Bi-LSTM f'_{bilstm} with Bidirectional Long-Short Term Memory (Bi-LSTM) units [60]. In both recurrent models, the recurrent unit appears after the convolutional block. A FC layer named the Regression Block is at each model's end. Fig. 2.4 Presents the scheme of proposed neural network architectures used in the following work.

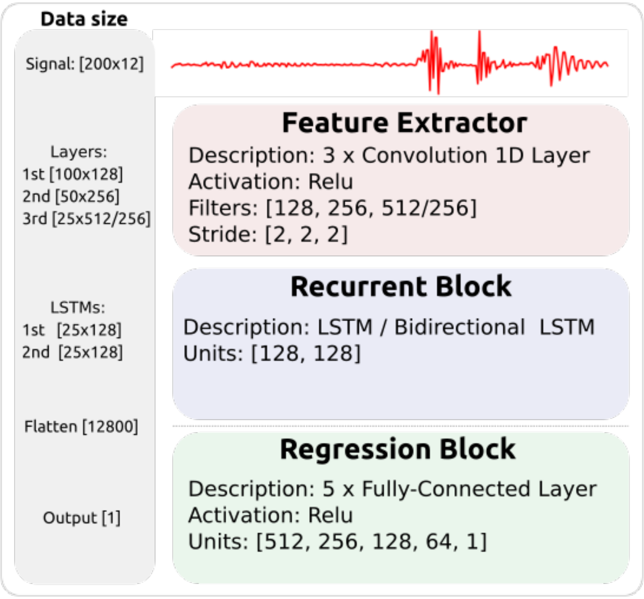


Figure 2.4: The Feature Extractor produces features using 1D convolutions. In the f'_{lstm} and the f'_{bilstm}, the Recurrent Block process these features to find relevant connections for the stiffness estimation. Finally, the Regression Block outputs a stiffness coefficient $k \in \mathcal{K}$.

Feature Extractor This module is responsible for extracting features from the raw signals while maintaining its output in the time domain. The input signal was a standardized sensor reading of 200 m samples consisting of 2×6 measurements. Hence, data could be further processed recurrently or passed to the Regression Block directly. The Feature Extractor consisted of 3 consecutive 1D convolution layers with strides equal to 2. The number of filters in the layers of the CNN was 128, 256, 512, while in the CNN-LSTM and CNN-Bi-LSTM models, the last convolutional block was reduced to 256 filters and replaced by the recurrent block of the same size.

Recurrent Block It processed high dimensional time series from the Feature Extractor in a recurrent manner using LSTM or Bi-LSTM in f'_{lstm} and f'_{bilstm} respectively. A resulting, fixed-length vector representing an entire sensor reading maps the input from the time domain to the feature space. Each recurrent cell consists of 128 units, as it was shown in Fig. 2.5. In the CNN-LSTM, both LSTM cells are organized in two sequential layers processing an input signal from the beginning to the end. Output features from that layer fed the Regression Block.

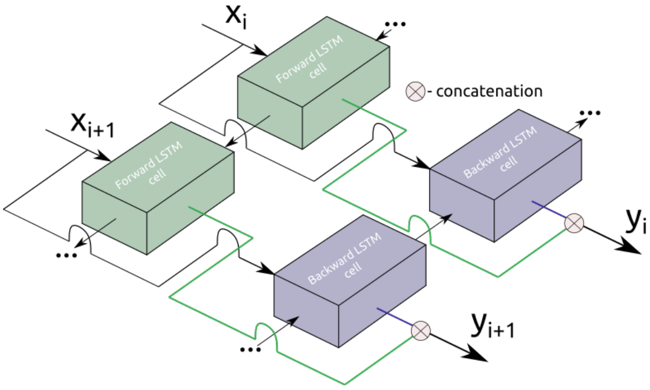


Figure 2.5: The core idea behind the Bi-LSTM used in the CNN-Bi-LSTM is as follows – to prevent losing a context by the cell, process a sequence from the beginning to the end, do the same in the reversed direction, and concatenate both passes. Input x_i refers to the i-th feature vector the convolutional block returns.

Regression Block It was the last global module used in tested architectures. It was responsible for a final estimation of a stiffness coefficient k. Using a FC layers is necessary because extracted features and time dependencies between them are critical ingredients in the regression process, but they are not the answer itself. Finally, it is necessary to transform obtained features into a stiffness coefficient k, made by the stacked FC layers. The number of units in each layer remained unchanged for all tested architectures and was 512, 256, 128, 64, 1.

2.4.5 Discussion on chosen methods

The proposed models for stiffness estimation based on time series of accelerations and angular velocities from the IMU mounted to the gripper consist of a Feature Extractor, Recurrent Block, and Regression Block. The first model uses only the Feature Extractor and the Regression Block, which is a pure CNN based on 1D convolutions with FC layers at the top. The second model adds an LSTM cell as a Recurrent Block to the pure CNN model, and the third model utilizes a Bi-LSTM as a Recurrent Block.

The pure CNN model has the advantage of being relatively simple to implement and train, and it can capture the local features of the time series well.

However, it may struggle with capturing the long-term dependencies between the data points, which can be important in stiffness estimation. The CNN-LSTM model can capture both local and global features by using the LSTM as a Recurrent Block. It can handle long-term dependencies and is more suitable for time series data than the pure CNN. However, it may be more complex to implement and computationally expensive. Finally, the bidirectional CNN-Bi-LSTM model has the added benefit of using a bidirectional LSTM that can capture information from both past and future time steps. This makes it suitable for modeling complex temporal dependencies and potentially results in even higher accuracy in stiffness estimation. However, it is the most complex model of the three and may require more computational resources to train and evaluate.

2.5 Unsupervised haptic recognition

2.5.1 Problem formulation

Let the $f_e \colon \mathcal{S} \mapsto \mathcal{Z}$ be the encoding function that inputs raw time signals from the multidimensional domain $\mathcal{S} \colon \{s \in \mathbb{R}^n\}$ (e.g., 3-axis forces). Then transforms them to the latent space $\mathcal{Z}$, and $f_d \colon \mathcal{Z} \mapsto \mathcal{S}$ be a decoding function that maps these features to the original domain again. Let the clustering problem be a grouping of n points, such that $s_i \in \mathcal{S}_{i=1}^{n}$ into p clusters, where each cluster is associated with a centroid u_j, where $j = 1, 2, ..., p$. Given the theoretical formulation above following could be introduced:

- the baseline of learning-based algorithms for clustering unlabeled data using evaluation metrics presented in subsection 2.5.4. If the input data was a time signal, f_e for baseline methods was a well-known dimensionality reduction method Truncated Singular Value Decomposition (SVD);

- the author's implementation[1] that follows [5]. It uses a gradient descent, a mean-square error loss function l_{MSE}, and divergence loss l_{KL} to cluster incoming data into a predefined number of bins. The presented method can work on raw signals without f_d and f_e. However, optionally, we can also use the latent representation, which approximates the encoding-decoding function $\hat{x}_i = f_d(f_e(x_i))$ to represent signals in the latent space, and in that space, do the clustering. That method and baseline algorithms are described further in subsection 2.5.5. The distance between

[1] https://github.com/mbed92/haptic-unsupervised

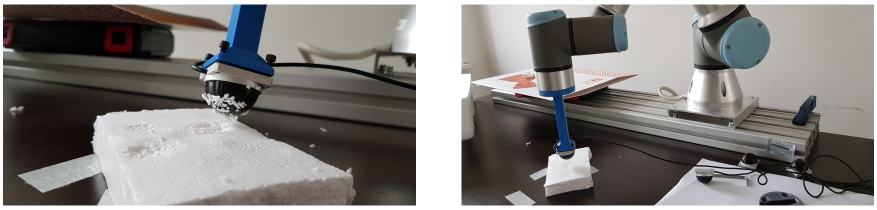
raw signals or latent vectors in the $\mathcal{Z}$ and centroids u in the form of an auxiliary Student's t-distribution used to calculate the Kullback - Leibler (KL) divergence loss l_{KL} that minimizes these distances during a training phase;

- the $\mathcal{D}_{touch}$ dataset of pairs $d_{touch}\colon (s, c)$ consists of real-world 3D force readings gathered while touching different materials using a robotic manipulator associated with one of ten semantic classes. That dataset was created by the author of the following thesis together with other authors of [61]. It is worth noting that, in the following experiments, there was no train/validation/test split because in unsupervised learning, we do not have access to labels, so there was no clear definition of how that split would look like. Additionally, real-world datasets typically suffer from a limited number of samples due to time and hardware constraints, as was the case here. Therefore, the author of the following dissertation resigned from splitting the dataset and focused on analyzing the results from all available samples;

- the BioTac Grasp Stability Dataset V2 (BiGS V2) [62] dataset $\mathcal{D}_{biotac}$ of pairs $d_{biotac}\colon [(s, c)]$ that consists of electrodes readings gathered at the same time from the BioTac sensors mounted on a hand-like gripper while grasping one of 51 classes of objects. That dataset is a follow-up work of [2] that extends the previously used BiGS dataset. Consistent with the Touching dataset, there was no train/validation/test split;

2.5.2 Touching dataset

Fig. 2.6a presents the 3-axis optical force sensor OptoForce used to create the Touching dataset. This sensor has a diameter of 32 mm, weighs 30 g, and its shapes resemble a human fingertip. It can measure forces by utilizing optical principles. In the central part of the sensor, there is an infrared emitter with four receivers. The semi-spherical rubber internal surface is layered with a mirroring substance. Hence, reflections of infrared rays are highly dependent on the deformation of this hemisphere. Force measurements have three dimensions, and the sensor's precision reaches 6.25mN. The OptoForce sensor has a nominal force capacity of 100N on the Z-axis and 50N on the X and Y-axis, and it can be overloaded on each axis by 200%. Again, we used the collaborative robot UR3 to gather dataset samples. It can estimate forces and measure torques in joints, enabling us to perform touching movements by pressing with a maximum force of 25N on the Z-axis of the robot's base coordinate system. The overall setup is shown in Fig. 2.6b.



(a) Touching Styrofoam.

(b) Overview of the setup.

Figure 2.6: The Figure presents robots and sensors used to create the touching datasets. The dataset focused on touching a variety of materials was created using a UR3 robot (Fig. 2.6b) with an OptoForce sensor mounted on the top of the end-effector that is visible in Fig. 2.6a.

The Touching dataset consists of 1293 labeled signals. Each sample consists of a 3-axis signal of forces recorded while touching the material with a length of 190 samples. The dataset includes ten classes of popular materials varying in thickness, stiffness, and texture. Items included are everyday objects, such as plane cardboard, corrugated cardboard, rubber, leather, linen bag, plastic plate, metal sheet, sponge, styrofoam, and a plastic bag. Sensors in experiments measure forces directly, allowing not only for the classification of materials but also to work on the regression, i.e., estimation of physical parameters of the material. Using other sensors used for material classification from the contact, such as accelerometers [63] or microphones [64], there is no possibility of estimating such parameters.

2.5.3 BioTac Grasp Stability Dataset V2 (BiGS V2)

This dataset is an extended version of BiGS [2] (used and described in other experiments of the following thesis regarding multi-modal fusion 4.4.2). The dataset follows the same structure as the original one and includes signals from it but adds 10 new classes, yielding possible benefits in any learning-based approach, especially unsupervised learning. That version consists of 5831 raw electrode readings recorded while grasping 51 different objects' categories. Each data sample is a measurement from three BioTac sensors mounted on a hand-like gripper. Three sensors on fingers are used to grasp objects, and each sensor returns a signal from 24 electrodes, giving 72 measurements per grasping episode. More information available in [62] and the corresponding repository[2].

[2] https://github.com/3dperceptionlab/biotacsp-stability-set-v2

2.5.4 Evaluation metrics

Clustering accuracy In the unsupervised setup, the accuracy might be obtained by solving a linear sum assignment problem [65] using a cost matrix D of a bipartite graph. The $k \times k$ matrix D includes assignments of each supervised class into unsupervised clusters. The metric achieves 100% when all categories are separate clusters. In all algorithms to calculate the metric, the expected number of clusters was preset to the number of classes used at the training phase. However, it does not mean that the model will categorize data into the same number of clusters.

Mutual information The following metric [66] shows the similarity between two clustering strategies in terms of the amount of surprise (or, in other words, uncertainty) in the results and mutual dependence between two random variables. It is typically used to evaluate the clustering quality or identify the most relevant features in a dataset. The non-normalized score ranges from zero to infinity, where higher values indicate more pure clusters. However, in the experiments, the normalized score was used. Similarly to clustering accuracy, it expects actual labels. Thus, it could be used with the desired number of clusters equal to the valid number of classes in the dataset.

Purity It shows to what degree each set contains one type (class) of samples – how pure clusters are. Each group is assigned to the class most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned samples and dividing by the number of samples. Bad clustering has purity values close to 0. A perfect clustering has a purity of 1. High purity is easy to achieve when the number of clusters is significant - in particular, purity is 1 if each sample gets its cluster. Thus, we cannot use purity to trade off the clustering quality against the number of groups.

2.5.5 Adapted methods

Embedding clustering Deep Embeddings Clustering (DEC) [5] is a relatively new method that uses deep learning to learn feature representations of data, which are then used for clustering. This approach has several advantages over classical clustering algorithms from the Scikit-learn package. DEC can learn complex non-linear relationships in the data, whereas classical algorithms are limited to linear relationships. DEC allows deep embedding clustering, which produces more accurate and meaningful cluster assignments. Moreover, it is computationally efficient. It can handle large datasets and does not require extensive hyperparameter tuning, which can be time-consuming with classical algorithms. Overall, DEC is a powerful and efficient method for clustering data with many advantages over classical ones.

Ward The Ward algorithm is a hierarchical clustering method used to group objects based on their distance from one another. It works by iterative merging clusters to minimize the distance between the combined sets. This distance is typically measured using the Euclidean distance, the straight-line distance between two points. The algorithm continues to merge clusters until all objects are a single cluster. The result is a hierarchical tree or dendrogram that can be used to visualize the groupings of the entities.

Spectral It groups data points based on the eigenvectors and eigenvalues of a similarity matrix constructed based on the distance-based similarity between data points [67]. Eigenvectors and eigenvalues create a new feature space in which we look for clusters using other algorithms. The K-Means method was the final clustering strategy for these embeddings in the following experiments. However, it can be computationally expensive and sensitive to the choice of similarity measure. In the following experiments, the Spectral Clustering method is an improvement of the K-Means method with more sophisticated data preprocessing – eigendecomposition reduces the number of features in the input data while preserving differences between data samples.

K-Means One of the most popular clustering algorithms used nowadays [68]. It scales well to large datasets and has been used across an extensive range of application areas in many fields. K stands for the number of disjoint clusters in the data we expect. *Means* refer to so-called centroids - the cluster centers we want to find. Finding sets is an iterative process of calculating distances from all data points to (initially random) centroids, classifying each sample based on the closest distance, and then recomputing centroids' positions by taking the mean of each created group. Generally, K-Means is a good starting point for clustering analysis because of its simplicity and speed (linear complexity). However, it is fragile to initial random choice of the centroids, and cluster shapes are considered elliptical, which is probably not valid in real-world data.

Gaussian Mixture Model (GMM) It is a probabilistic model that assumes that the data points follow a mixture of several multivariate normal distributions, each representing a cluster. The parameters of the GMM, such as the mean, covariance, and mixing coefficients of the Gaussian distributions, are estimated from the data using the Expectation-Maximization (EM) algorithm. The EM algorithm iteratively improves the estimates of the parameters until convergence, at which the data points belong to clusters based on the probabilities of each Gaussian distribution. The EM method is flexible and can handle non-spherical sets. Still, it can be sensitive to the initialization of the parameters.

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) The method [69] uses a hierarchical approach to identify clusters in large datasets. It starts by constructing an in-memory data structure called the Clustering Feature tree, which stores the feature vectors of the data points and summary statistics about the feature vectors. The Clustering Feature tree is constructed iteratively by inserting the feature vectors into the tree and merging similar nodes. Once we have the Clustering Feature tree, the clusters are identified by cutting the tree at a specific height, which determines the granularity of the groups. BIRCH is efficient and can handle large datasets, but it is sensitive to the choice of parameters, such as the maximum number of nodes in the Clustering Feature tree and the threshold for merging nodes.

Agglomerative A hierarchical clustering method works by splitting (top-down) or merging (bottom-up) data into groups and clustering by revealing a hierarchy presented originally in [70]. The following experiments utilized the bottom-up approach. This method starts by treating each data point as a separate cluster, then iteratively merges the most similar clusters together based on the average distance between their points. It reveals the hierarchy in the data by a so-called dendrogram that can be cut at different heights to form a desired number of clusters. Unlike other algorithms, it allows for the exploration of clusters at different levels of granularity. It is insensitive to the type of similarity measure used, but it brings a significant computational burden.

2.5.6 Discussion on chosen methods

DEC seems to be the best candidate for the specified task that has shown great potential in discovering meaningful structures in complex data such as images, text, and time series. Compared to other traditional clustering algorithms, DEC leverages the representation learning capabilities of deep neural networks to learn feature representations in an unsupervised manner automatically. Efficient representation learning lets the model identify underlying patterns and structures that may not be immediately apparent in the raw input data. Additionally, DEC is known to be more scalable and efficient than other methods, especially when dealing with high-dimensional data. Traditional methods rely on assumptions that could be unsuitable for such a complex task as unsupervised haptic recognition. GMM assumes that the data follows the Gaussian distribution, which might not be accurate in real-world problems. Hierarchical algorithms like BIRCH, Ward, and Agglomerative clustering are sensitive to noise and outliers and may produce inconsistent results for datasets with high variability or irregularities. K-Means assumes that clusters are spherical and equally sized, which makes it inefficient in capturing complex and irregular shapes, as might be the case in haptic datasets. Finally, Spectral clustering may not be effective in identifying clusters with varying densities, as it relies on graph partitioning techniques that may need to handle such scenarios more effectively. DEC appears to overcome all the limitations of traditional methods. However, all methods tackle the clustering problem differently, and a thorough evaluation of the performance of all methods is necessary to ensure the best choice for the specific task.

Chapter 3

Terrain recognition

This chapter and the corresponding results in Part III incorporate material from the following publications.

- (© 2021 IEEE) M. Bednarek, M. Łysakowski and J. Bednarek, M. R. Nowicki, K. Walas (2021) "Fast Haptic Terrain Classification for Legged Robots Using Transformer". In: European Conference on Mobile Robots (ECMR)
- (© 2022 ELSEVIER) M. Bednarek, M. R. Nowicki, K. Walas (2022) "HAPTR2: Improved Haptic Transformer for Legged Robots' Terrain Classification". In: Robots and Autonomous Systems – Selected papers from the 10th European Conference on Mobile Robots

3.1 Introduction

The mobility of legged robots is a vital factor that gives them an advantage over wheeled platforms, which struggle to function in human-made domains with non-flat areas. Autonomously traversing kilometer-scale, natural environments demand excellent walking capabilities, i.e., agility, robustness, and low-computational demand to avoid battery draining [71]. They must fulfill a broad spectrum of mission-specific requests, including natural caves exploration, urban environments inspection (e.g., sewers), or search and rescue tasks. One of the challenges of the real-world operation of legged robots is negotiating the unknown and unexplored terrain where the robot has to adapt its gait to the changing environmental circumstances. Planning and predicting the system's behavior in many scenarios is possible. However, there are cases when the robot has to react to unanticipated environmental measurements [72]. In [73], authors presented such an adaptive behavior describing a robot's movement as a terrain function.

The following Chapter predominantly concentrates on the terrain classification with haptic contact, a well-researched topic with most solutions focusing on F/T signals from the sensors mounted on the feet [74, 56, 75, 61]. These methods focus on obtaining the best results in accuracy measured on the registered dataset. However, the accuracy metric is only a part of the whole picture. A method should not achieve satisfactory classification only but also a short inference period. Moreover, computations should also be completed with limited processing resources as an autonomous walking robot has multiple nodes that must operate simultaneously to achieve desired outcomes in real-life challenges. Additionally, the following chapter argues that kinesthetic sensing, defined as sensing the position of a robot's limbs, leads to successful haptic perception. Typically, a sensor used to measure inertia in robotics is an IMU commonly mounted in a base of a walking machine. However, that placement is valid for kinesthesia because all haptic sensations, slopes, and terrain types must influence that sensor's readings.

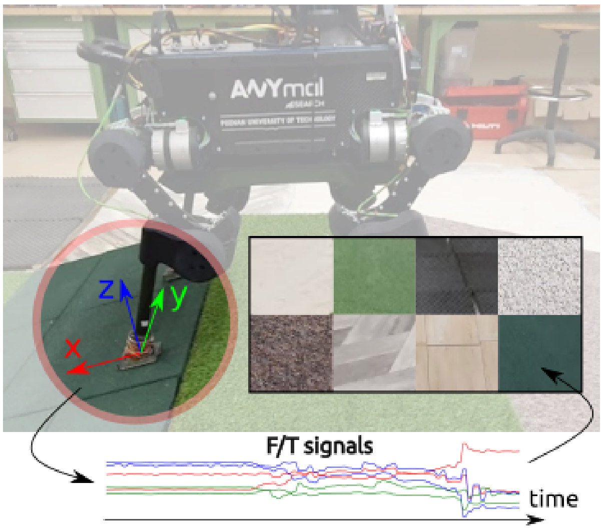


Figure 3.1: (© 2021 ELSEVIER) In the picture, the ANYmal robot collects haptic F/T measurements with its compliant feet when walking on diverse terrains. Terrain classification is paramount for gait adaptation to ensure stability. Moreover, there is a need to achieve a short inference period with resource-constrained processing units. Therefore, the transformer-based HAPTR and HAPTR2 methods were proposed.

The following Chapter presents the deep learning model called HAPTR with further extensions (HAPTR2). Provided tests include a thorough evaluation of the HAPTR model targeting previously omitted aspects like inference time and the method's robustness. This work contains a baseline in the results obtained with three adapted methods ranging from classical machine learning approaches to the CNN. Moreover, the chapter provides results from other state-of-the-art solutions whenever possible and compares the accuracy on two publicly available datasets. Finally, an ablation study was employed to tackle the problem of the robustness of the proposed perception system.

Therefore, the contribution presented in the following chapter comprises four items. Firstly, the work focuses primarily on transformer-based deep learning models for terrain classification – HAPTR and HAPTR2. The lightweight implementation focused on short inference times and robustness against changes in input signals. Secondly, the research was conducted towards the MAL as a way to increase the robustness of terrain classification in real-world scenarios. Then, the experiments include a benchmark of terrain classification systems from the perspective of autonomy requirements. It includes the comparison of accuracy, inference time, and robustness against data degradation. Eventually, a transparent evaluation of the proposed and several adapted methods was included based on two publicly available terrain classification datasets to create a baseline for further works to compare and overcome the proposed solution. All the code and datasets to reproduce the results are available online [1] publicly.

3.2 Related work

3.2.1 Terrain recognition for legged robots

In [76], authors introduced one of the first contemporary approaches to terrain recognition for walking robots, where they tackled the problem of blindly assessing terrain properties using only currents from motors and contact forces of quadrupedal robot legs. Then, the AdaBoost [77] algorithm processed extracted features from input signals, which revealed that a Ground Reaction Force (GRF) is one of the essential properties describing a terrain under a moving leg. In [78], authors proposed classifying terrain samples on tiny robots, where they directly measured a GRF from the designed miniature array of sensors. Direct measurements of the robot and ground substrate interaction were also described in [79]. A quadruped robot adjusted a Center of Gravity (CoG) based on a terrain class to accommodate terrain properties changes. The step further in using terrain classification is to allow the robot to adapt its gate to changing traction conditions [80]. Predicting the risk of collapse of the structures negotiated by the legged machine [81] is even further extension. Most

[1] https://github.com/mbed92/haptic_transformer

of the research focuses on different types of hard materials, but there are also soft substrates in the dataset used in the following work. In [82], the authors explicitly mentioned soft ground and provided the method of in situ measuring the terrain parameters. This work [83] and its further extension [84] presents a more analytical approach for the robots dealing with soft materials and gait adaptation. Terrain classification with other modalities like vision [85] or acoustic sensors [86] is feasible yet less accurate than direct force measurements. For terrain classification, the fusion of three different modalities, vision, depth, and touch, was described in [87].

Moreover, tactile data allowed for self-supervised visual terrain learning in [88]. In [74], authors analyzed tactile signals from impact motions of the quadrupedal robot's leg to classify different soil characteristics. In [61], authors proposed a RNN with convolutional blocks that achieved state-of-the-art results in the terrain recognition task. However, that method was able to work with fixed-length input data only. In [75], the masking mechanism for CNNs was proposed to manage variable-length signals. In [56], authors also used raw and variable-length data recorded during walking sessions on different robots. Due to the semi-supervised machine learning approach, their terrain classification method achieved better performance than frequency-domain classifiers and required fewer annotations.

3.2.2 Time series classification

Time-series data mining has become an emerging field over the past decades due to the increased availability of temporal data [89]. We can approach the problem by looking at the signal properties. Authors of [90] compared several similarity measures in the distance-based Time Series Classification (TSC), showing that no individual distance measure is significantly better than the Dynamic Time Warping (DTW) [91]. In the following experiments, a popular K Nearest Neighbour (KNN) classifier uses this distance measure to work on time-domain signals (KNN-DTW), which the community considers a stable baseline in the field. However, authors of [90] also proved that using a heterogeneous Elastic Ensemble (EE) will outperform the DTW even though any of the individual parts of EE were not achieving higher accuracy than DTW. Authors of the Collective of Transformation-Based Ensembles (COTE) algorithm [92] used such an ensemble of classifiers based on various feature spaces. The follow-up work – Hierarchical Voting (HIVE) (HIVE-COTE) [93] leveraged the performance by the hierarchical voting system, which resulted in superior performance among other algorithms. The HIVE-COTE is considered a state-of-the-art method for TSC.

Still, it exhibits a very high time-complexity due to many algorithms in an ensemble of classifiers. To illustrate that phenomenon, one of them – the Shapelet Transform [94] achieves the time complexity $O(n^2l^4)$, which makes HIVE-COTE impractical in many real-time scenarios [89]. Authors of [95] presented the Bag-of-SFA Smbols (BOSS) algorithm, which extracts words from input signals and learns to classify them in that space by measuring the frequency appearing of each word in a time series. An alternative method was presented in [96], where the decision tree forest partitions the data under chosen distance measure.

Recently, learning-based approaches started to play a significant role in the TSC. In the range of state-of-the-art methods, the RNNs with convolutional blocks, or Convolutional Neural Network (CNN)s, play a prominent role. In [97], authors proposed the InceptionTime classifier based on the Inception-v4 [98] architecture that achieved cutting-edge scalability with decreased training time. A similar approach, called the Random Convolutional Kernel Transform (ROCKET), was proposed in [99]. The authors showed that their model achieves state-of-the-art accuracy while maintaining a relatively short training time. It was possible by using multiple convolution kernels instead of stacking an ensemble of classifiers. In [100], authors presented the Temporal Convolutional Network (TCN) as a convolutional alternative for RNNs with a comprehensive comparison to the state-of-the-art recurrent models.

Despite the undeniable success of transformers [101] in natural language processing [102], image recognition [103] or object detection [104], no prior works used transformers in the terrain classification in the real-world scenario with a walking robot.

3.3 Proposed solution

Mobile robots' popularity stems from their design, enabling them to interact with the environment in a plethora of outdoor and unpredictable scenarios [105]. Because of their versatility, resistance to environmental changes, and adaptive behavior, more and more companies consider them for commercial usage [106, 107, 108, 109]. Therefore, in many industrial applications, terrain recognition is a significant challenge. Walking robots can perceive their surroundings using various tactile-based or vision-based methods. However, vision-based methods exhibit limited applicability, as they are highly vulnerable to lighting conditions, fog, snow, dust, occlusions, low-texture surfaces, and other outcomes that might appear when a walking machine enters some hazardous area. One can observe that the focus of the robotics community turned mostly into tactile-based methods that utilize F/T sensors [74, 110], IMU [56], and proprioception [87, 111].

The novel terrain classification method presented in this Chapter was primarily designed for fast and robust inference on a legged robot using raw sensory signals. The HAPTR basis on the popular transformer architecture presented in [101] and [103] formerly. The extensive experiments scope included an accuracy comparison on the Poznan University of Technology ANYmal dataset (PUTAny) [18] involving four different data-driven methods in multiple setups. Then, to verify our claims about the HAPTR, there was added the section about cross-validation on another state-of-the-art Queensland Civil and Administrative Tribunal dataset (QCAT) [112], which presents the comparison against results presented in [56]. Moreover, the work presents the novel MAL as a solution that increases the robustness of predictions against environmental impacts that might occur during a legged robot's operation. In all experiments, the HAPTR achieved classification accuracy on the same level as the best models included in the comparison, having over 30×less learnable parameters and maintaining a short inference time. Finally, ablation studies on the MAL were included by deeply investigating its influence on the perception system's robustness.

3.4 Terrain classification using Transformers

3.4.1 Problem formulation

Let the $f\colon \mathcal{S} \mapsto \mathcal{C}$ be the haptic classification function that assigns a discrete-valued class from the counter-domain $\mathcal{C}$ to the raw, multidimensional signal sample from the set $\mathcal{S}$. The set $\mathcal{S}\colon \{s \in R^n\}$ consists of n-dimensional sensory signals from the available robot's equipment, such as F/T sensors, or the IMU mounted in its base. Typically, in the research software, the number n is called the number of *axes* of the input signal. The set $\mathcal{C} = \{0, 1, 2, ..., c\}$ is integers, where c is the total number of terrain classes expected in the dataset. Additionally, let the $f_{mod}\colon \mathcal{S} \mapsto \mathcal{W}$ be the modality weighting function that associates each multi-modal time series s with a set of modality weights for each time step. The set $\mathcal{W}\colon \{w \in \mathbb{R}^{m \times t_{max}}\}$ includes floating-point numbers that correspond to the notion of *importance* of input modalities, where m is the total number of modalities, and t_{max} is the time-length of an input signal. For the purposes of the dissertation, let's define *modalities* as a set of axes associated with signal referring to the homogeneous physical value, e.g., 3-axis F/T signals come from 2 different modalities. However, forces and torques might not be considered different modalities in the strictly physical sense, as they come from the same sensor for the sake of conciseness. The function f_{mod} would associate to the F/T signal set of two weights for each time step, where weights at time t are defined as $w_t = \{w_F, w_T\}$.

Given the problem formulation above, the following components of this work could be introduced:

- an extensive comparison of classification methods in multiple experiments subsumed here under the common definition of $f' \colon \mathcal{S} \mapsto \mathcal{C}$ for the sake of clarity, which approximate the function f using deep neural networks or machine learning approaches;
- an attention module f_{mod} in the form of a differentiable layer that increases the robustness of a function f' against input data deterioration;

3.4.2 Terrain datasets

Poznan University of Technology ANYmal dataset (PUTAny) The following dataset is the set $\mathcal{S} \colon \{\mathbb{R}^6\}$ of 3-axis force and 3-axis torque data samples, where the ANYmal [6] robot was continuously walking on different real-world terrain samples with no additional exploratory moves. It had mounted F/T sensors on all feet. However, the dataset consists of samples representing signals from one foot only, so there was no information regarding which foot the signal was recorded from. Fig. 3.2 presents the map created with eight different terrain types. This dataset is the follow-up work of [75], but with the number of terrain classes $c = 8$ (previously six) and with several other improvements. Firstly, the robot had compliant, sensorized feet. Moreover, available terrains included also slopes to add samples at some inclination. Finally, all-terrain samples created a single walking area. The dataset is publicly available [2].

During the walking session, the ANYmal robot shown in Fig. 3.1 had sensorized, compliant feet [113] that consist of flat contact surfaces with a range of motion 50° for the pitch and 30° for the roll axis. The robot traversed flat surfaces and a ramp requiring the feet to adapt to the terrain type and shape. F/T sensors placed inside the feet were custom-made and can sense up to 1000 N in the Z direction (along the robot's leg), 400 N in the ground surface, and up to 10 Nm of torque in each axis at a frequency 400 Hz. In the dataset F/T, signals were cropped to $t_{max} = 160$ registered during the instant of contact.

In the following chapter, eight different terrains were used (Fig. 3.3): carpet, artificial grass, rubber, sand, foam, rocks, ceramic tiles, and Polivinyl Chloride

[2] https://drive.google.com/file/d/1QP-a1Y78LaKVN_mLt91b_1OT_5YDyVD_/view

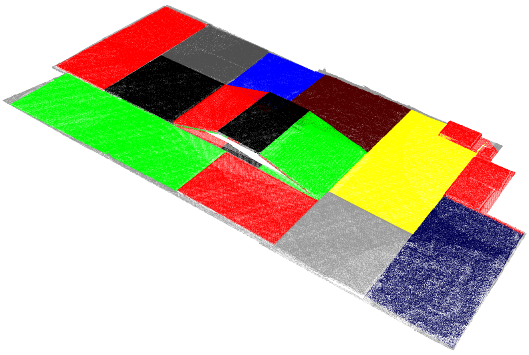


Figure 3.2: (© 2021 ELSEVIER) The map with eight terrain classes and a slope was used to register the PUTAny dataset. The ground truth map for data labeling was registered with a 3D laser scanner (SURPHASER 100HSX), while the walking robot's pose was determined with the OptiTrack system. Colors correspond to different classes: red – rubber, green – carpet, blue – PVC, black – artificial grass, yellow – ceramic tiles, brown – sand, dark blue – rocks, grey – foam.

(PVC). One can observe that the adaptive foot slopes differ depending on the terrain type, properties, and shape. The robot performed a statically stable gait with only one leg in a flight phase at a time. After a flight phase, all feet were on the ground. Then, a flight phase started all over again with a different leg.

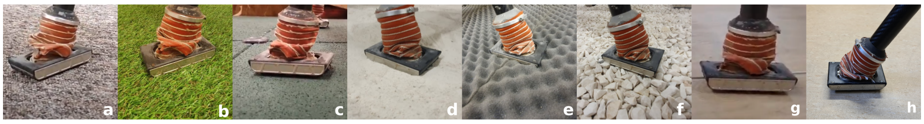


Figure 3.3: (© 2021 ELSEVIER) Terrain types included in the dataset: *carpet* (a), *artificial grass* (b), *rubber* (c), *sand* (d), *foam* (e), *rocks* (f), *ceramic tiles* (g), *PVC* (h). Each terrain gave unique F/T feedback enabling the classification of these samples.

The dataset consists of recorded terrain samples divided into 3443 training, 1148 validation, and 1148 test subsets. Each signal contains 160 3-axis force and 3-axis torque measurements. The dataset comes from the continuous walking session, which is why it consists of an uneven number of samples registered for different classes (Fig. 3.4 presents their distribution). In the dataset, there is no difference between samples for each foot. All samples have equal t_{max}.

Queensland Civil and Administrative Tribunal dataset (QCAT) Another state-of-the-art dataset was employed to measure the performance of HAPTR models – the QCAT dataset. It consists of recordings from multiple walking sessions of the quadrupedal, self-configurable robot called DyRET [114].

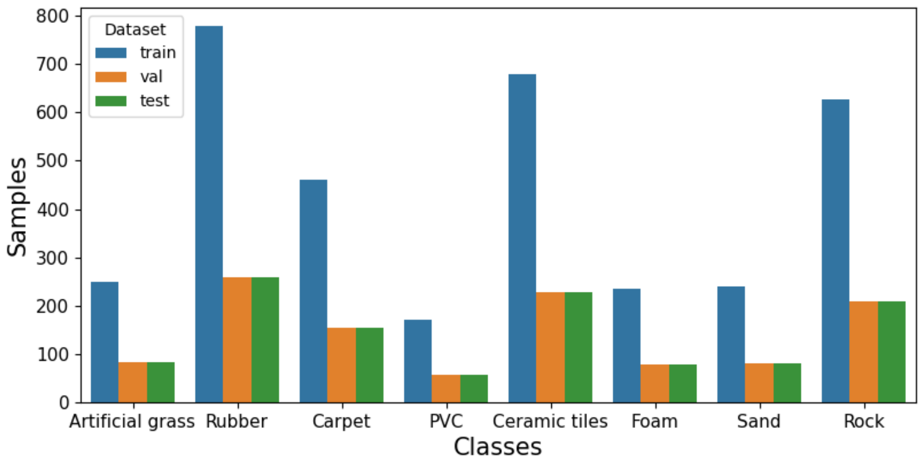


Figure 3.4: (© 2021 ELSEVIER) The distribution of terrain samples for the PUTAny dataset.

Raw signals come from the IMU mounted in its base and 3-axis forces from spherical sensors mounted at the tips of the robot's legs. DyRET [114] is a four-legged robot with a dynamic morphology, designed to adapt the lengths of its limbs to different terrains. The kinematic chain of its leg was composed of two rotational joints intended for locomotion, and two, slow-changing prismatic joints for elongating and shortening the leg. It had the Xsens MTI-30 IMU mounted in its base consisting of a 3-axis gyroscope, a 3-axis accelerometer, and a 3-axis magnetometer. Each foot has the 3-axis force sensor Optoforce OMD-20-SH-80N.

The QCAT dataset consists of 2880 force and IMU samples – 6 terrains × 10 trials × 6 speeds × 8 steps. Each force and IMU time series had $n = 22$ axes – 4 × 3-axis force sensor and 2 × 3-axis angular velocities/linear accelerations, and 4-axis quaternion representing base's orientation. In the experimental section with the attention module, those axes created two modalities representing data from homogeneous classes of sensors – force and IMU separately. Each signal had $t_{max} = 662$. Similar to the PUTAny dataset, it does not differentiate feet or evaluate performance for each foot separately. The QCAT dataset is publicly available [3].

3.4.3 Adapted methods included in the comparison

Convolutional neural network with recurrent modules (CNN-RNN)
In [75], authors tackled the problem of a terrain classification for the legged

[3] https://data.csiro.au/collection/csiro:46885v2

robots. In their experiments, they verified two deep learning models (RNN and CNN), a well-established SVM [115] as a baseline, and the FC network working on Fast Fourier Transform (FFT) features extracted from raw signals. They compared all of the mentioned models on the fixed-length and variable-length time series. Both deep learning models and CNN performed better than the traditional SVM method. However, this does not necessarily imply that they are faultless. Long sequences of F/T signals caused a well-known problem of gradient vanishing in the RNN, resulting in significantly deteriorating classification accuracy. CNN was free of this shortcoming, but using 1D convolutional layers forces working only on fixed-length input signals. Eventually, the feature-engineering solution based on CNN required calculating the descriptors of input signals, thus, not achieving satisfactory accuracy. The most recent version of this model described in [18] is the composition of all three components described before – RNN, CNN, and the MLP at the top. RNN and CNN components process variable-length data. When it comes to the RNN, it is a natural consequence of the recurrence, but CNN uses an additional masking mechanism. It pads variable-length input signals with zeros to match the fixed length. This model achieved the highest accuracy among all tested methods.

Dynamic Time Warping K Nearest Neighbors (KNN-DTW) The KNN-DTW [90] was a baseline in the experimental section about the terrain classification. It is a well-established, distance-based classifier that uses the DTW algorithm to measure the similarity between query signals (input time series) to the database of signals created during the training. While classifying, it determines the three closest matches (k=3) to the query signal by matching a training sample to the whole database. A training sample is then classified if the majority (at least 2) of matched sequences from the database represent the same terrain class. The experimental section includes results gathered using an efficient implementation of the KNN-DTW in Python available in Sktime [116] library.

Random Convolutional Kernel Transform (ROCKET) The authors of the ROCKET algorithm proposed a method for the TSC that works with a low computational burden while being invariant to the representation of the input features (e.g., a shape or a frequency) because it uses a single, general mechanism – a convolution. Firstly, some significant number of random convolutional kernels transform input time series into a feature space. The authors claim that these features – in combination – capture relevant information for the TSC. Finally, any classifier might use them for training. However, the authors recommend linear classifiers (e.g., a logistic or Ridge) because they can utilize a small fraction of information from all input features. Unlike artificial neural networks, kernels' weights are not learnable but randomized. Hence, the computational cost of training is low. The initial design was to work primarily with univariate signals. In the experimental section, each axis of F/T signal was transformed separately to a feature space using 10000 random kernels. Once in a feature space, different information channels were concatenated to form one feature sequence passed to the final classifier.

Temporal Convolutional Network (TCN) In [100] authors proposed a general, convolutional-based framework called the Temporal Convolutional Network for time series processing. It fits the requirements established in this work and handles longer sequences than recurrent neural networks. The TCN consists of multiple 1D fully-convolutional layers, where all subsequent hidden layers are in the same length as previous ones. It keeps the output of the TCN the same length as the input. The *casual convolution* is a mechanism added to keep the past time samples away from the current ones, so there is no leakage from the future into the past. The more time steps are in the input signal, the more complexity it adds to the TCN. To overcome this shortcoming, the authors proposed to use dilated convolutions with the exponential receptive field in the consecutive layers. This procedure allows handling significantly longer sequences while achieving a relatively low computational burden. The TCN does not contain any form of memory or recurrence while providing state-of-the-art results. The original TCN returns a features sequence, which does not apply to the task of terrain classification. Therefore, the adapted model includes the Multi-Layer Perceptron (MLP) layer at the top to predict terrain classes. In the experimental section there were evaluated several configurations of the TCN that differ when it comes to the number of convolution levels (LE), hidden units per each level (HI), and the number of hidden neurons in the (MLP):

- *Light* - LE=4, HI=8, MLP=128,
- *Base* - LE=8, HI=16, MLP=256,
- *Large* - LE=16, HI=25, MLP=256.

The authors of [100] shared the implementation of their TCN, which was a basis for the experimental verification in the following work.

3.4.4 Contributed methods based on transformers

> The following section presents two methods in chronological order, as they appeared in the publications of the author of this thesis. They work under the same principles. However, the second one was the follow-up work that included an improved version of the HAPTR with the MAL.

Haptic Transformer HAPTR The most recent peak in popularity of transformers, specifically their application to computer vision problems [104], was a foundation under consideration of this approach for terrain classification. No previous methods utilized transformer architecture to process multi-modal, raw sensory feedback to classify terrains. However, when writing this dissertation, it became apparent that other research groups were at the same time working on the same subject but in different research domains. The time series transformer appeared, e.g., in [117] – the submission of this article took place one month after the author's initial HAPTR article [118], which indicates the high priority in the research community of the issue addressed. The HAPTR [118] comprises the first proposition to tackle the problem using such an architecture. It uses a self-attention mechanism instead of convolutions to classify the terrain. As the authors of [119] point out, the usage of multi-head attention modules improves the accuracy and generalization ability of the method by turning the training to domain-specific knowledge instead of data-specific (as is the case using convolutions), which would be beneficial in the robotic application. Similar to Vision Transformer (ViT) used in [103], the HAPTR uses a learnable linear projection layer to map a signal $s \in \mathbb{R}^6$ (e.g., a 3-axis force and 3-axis torque) into a sequence of the same length t_{max} in feature space of 16 axes. In the following work, samples of that sequence are called patches. Then, positional encoding (PE) is added to every patch to retain position information and passed to the Transformer Encoder Layer. Eventually, every vector is reduced in dimensionality before the final classification with the MLP. The implementation of the HAPTR is based primarily on PyTorch modules, especially TransformerEncoder and TransformerEncoderLayer.

For the HAPTR, there were evaluated variants with a different number of encoder's layers (L) and attention heads (H). Models are referred as:

- *Light* - L=2, H=4,
- *Base* - L=4, H=8,
- *Large* - L=8, H=8.

Improved Haptic Transformer with the MAL The HAPTR is the first attempt to tackle the problem of terrain classification using a transformer-based neural network. With a piece of domain knowledge about the problem itself, the HAPTR evolved to an improved version called the HAPTR2 as presented in Fig. 3.5.

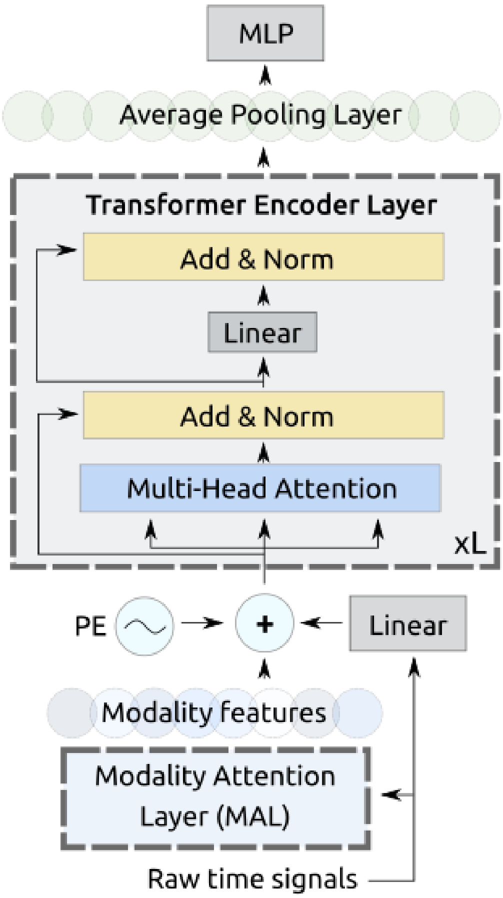


Figure 3.5: (© 2022 ELSEVIER) Improved HAPTR is the follow-up model based on [118] with MAL.

The main novelty of the presented method is the MAL, which basis on the Multi-Head Attention mechanism introduced in the [101]. In [119], the authors show that the composition of the attention mechanism with convolutional layers is complementary and generally beneficial. The former behaves like a low-pass filter due to the weighting mechanism presented in the Eq. 3.1, which flattens the feature maps. On the other hand, the convolutions are prone to capture high-frequency features and focus on data-specific knowledge (e.g., spikes, slopes, mean and variance changes). This phenomenon suggests that using the multi-head attention mechanism would improve the robustness of the robotic system, which might undergo a plethora of data-degradation scenarios in the real world. The MAL implements the $f_{mod}\colon \mathcal{S} \mapsto \mathcal{W}$ function and assigns importance weights w to the input modalities at each time step t. Firstly, input time series s is split according to its modalities (i.e., 3-axis force and 3-axis torque signals measured at the robot feet are separate modalities) and passed to 1D convolutional layers, which creates flattened representations of the multi-axes, modality signal representations of the same length t_{max} as inputs. Learnable linear layers process and shape each modality representation to so-called queries (Q), keys (K), and values (V) for the dot product attention layer. There are as many queries as input samples. Then, each sample is weighted between existing modalities and scaled by the factor of $(1/\sqrt{d_k})$, where d_k is the dimensionality of multiplied queries and keys. Therefore, the Eq. 3.1 describes activation of the MAL layer:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V. \tag{3.1}$$

The keys are formed of a matrix with size $d_k \times d_k$. Rows represent a $d - th$ modality with $d = 1, 2, ..., d_k$. Therefore, the closer the query is to the corresponding key, the higher weight is associated with that modality. MAL can work with any number of user-defined modalities, but in the scenario with forces and torques, they are two separate modalities. The d_k is equal to 2, and keys are composed of the 2×2 matrix. Finally, a softmax is applied on the scaled dot-product to obtain a probability distribution. Fig. 3.6 presents the information flow of MAL. Typically in transformer models, the self-attention layer delivers weights between all time steps, e.g., in Natural Language Processing, to reveal the contextual associations between the first and other words in the sentence. Nevertheless, in the following work, the attention layer discovers weights between time steps and entire modalities rather than between all pairs of time steps in the input signal.

Apart from the MAL, a complete list of changes introduced in HAPTR2 in comparison to the initial version includes:

- used an improved learning rate scheduling [120],
- an output from the MAL was concatenated with original input signals by channels axis,
- an average pooling layer replaced a mean operation before an MLP classification layer,

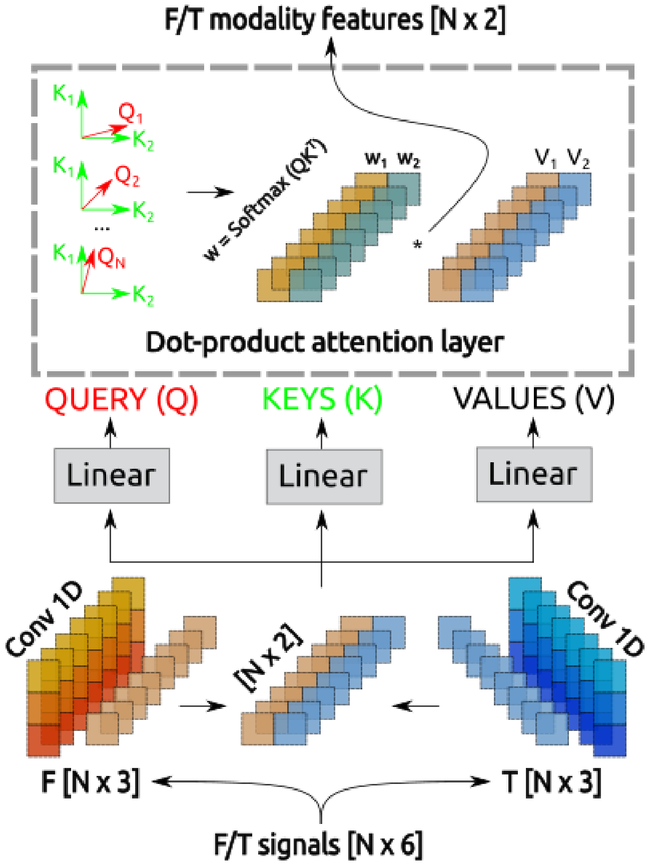


Figure 3.6: (© 2022 ELSEVIER) The visual explanation of the MAL used in the experiments. It was used to increase the robustness of the HAPTR model.

- included a batch normalization layer in the final MLP classification layer.

3.4.5 Discussion on chosen methods

The section discusses terrain classification using haptic signals and proposes four adapted methods for further experimentation: CNN-RNN, KNN-DTW, ROCKET, and TCN. CNN-RNN leverages the power of convolutional and recurrent neural networks to learn spatiotemporal features from high-dimensional haptic signals efficiently. This combination has shown great potential in similar tasks. However, one of the main challenges might be overfitting and a lack of interpretability of learned features. Moreover, they have limited usability when the sequence is variable-sized (however, that aspect was not considered in this study). A distance-based method such as KNN-DTW can show potential in handling time series and are generally easy to implement. On the other hand, they are computationally expensive, especially for large datasets, which can exclude them from real-world applications for terrain classification. ROCKET was implemented to be fast and straightforward. However, because of using convolutional kernels ROCKET will suffer from the same limitations. Finally, TCN appears to be a perfect choice for terrain classification of haptic signals as it leverages the advantages of CNN with the lack of disadvantages of recurrent modules of RNN. However, it might require careful hyperparameter tuning and a large amount of training data to avoid overfitting.

In the following experiments, there was presented HAPTR – a CNN model with attention layers that will possibly overcome the limitations of other methods by selectively focusing on the most informative parts of the input. Attention mechanisms allow the model to learn to selectively attend to specific parts of the input data based on their relevance to the task at hand. This selective focus can help to reduce noise and irrelevant information and improve the model's ability to classify complex data such as haptic signals. Additionally, by focusing on the most informative parts of the input, the attention mechanism can help to overcome limitations such as overfitting and improve the interpretability of learned features. In the following thesis, the author proposed to use additionally the Modality Attention Layer (MAL) that pushes further the ability to select relevant channels of the multi-dimensional signals and increase the interpretability of decisions made by the model.

Chapter 4

Robust multi-modal fusion

> The following Chapter and the corresponding results in Part 7 basis on the following publication of the author of this thesis.
>
> - M. Bednarek, P. Kicki, K. Walas (2020) "On Robustness of Multi-Modal Fusion – Robotics Perspective". In: Electronics Journal

4.1 Introduction

The multi-modal information fusion remains crucial in a wide range of robotic applications that require understanding various physical properties, like the texture of an object, its color, softness, and more. Due to the complex behavior needed to complete such a task, a coalescence of visual and haptic stimuli might be beneficial in, e.g., dexterous manipulation of rods. In the classical work [121], the authors observed relations between information from different human senses, which caused a multi-sensory illusion further called the McGurk effect. In robotics, various probabilistic models based on Bayesian inference are distinctive to multi-modal data fusion. However, commonplace methods might not be adequate due to many available multi-modal and multi-relational datasets. Deep learning methods typically manage such extensive size datasets with great success; however, everything comes with a price, which in this example is learning data contamination, biases, and a lack of explainability. In recent years, there has been a lot of research on efficient data fusion using machine learning, especially using deep neural networks [122]. Nevertheless, researchers focused on the improvements in the accuracy of their models and paid almost no attention to their robustness to non-nominal conditions, which are ubiquitous in robotics tasks.

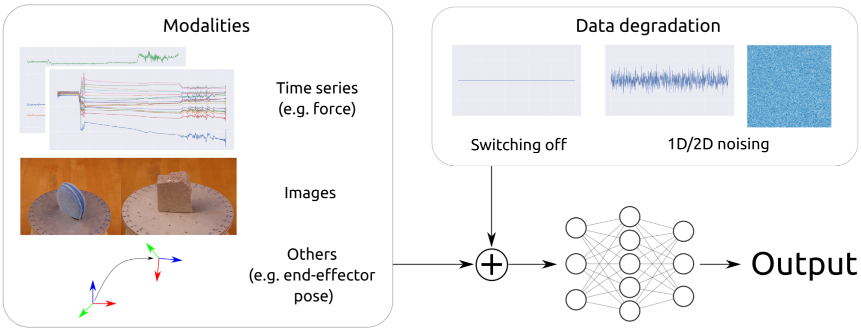


Figure 4.1: Fig. presents a general setup in conducted experiments. Different modalities subjected to different data degradation procedures were fed into the tested learning-based methods to evaluate their performance and robustness.

The multimodal fusion in dexterous manipulation [123, 124] nowadays is an emerging field. However, the development of the field for such complicated robotic applications very often depends more on advances in sensors, such as skin-like measurement devices [125, 126, 29] than fusion algorithms themselves. The emphasis on research of multi-modal systems in the field of robotics is especially visible in areas like image segmentation [127, 128, 129], 3D reconstruction [130] and a tactile understanding [4].

From the robotics perspective, the robustness of the multi-modal perception system against data corruption is of paramount importance. It can be achieved by finding interchangeable and complementary portions of information from different modalities in the input data stream. However, researchers mainly concentrated on establishing a benchmark in solved tasks rather than exploiting this complementarity. In state-of-the-art, there are very few examples of works where authors consider such interchangeability of modalities in the context of robustness, typical for real-life scenarios, noises, and sensor faults.

Multi-modal machine learning constitutes a scientific field of growing interest that brings many challenges. In [122], the authors have listed open questions that should find answers to advance state-of-the-art development. Firstly, how to represent the data (representation)? Then, how do we map knowledge from one modality to another (translation)? How do we find dependencies between heterogeneous modalities (alignment)? How do we join the multi-modal data stream together (fusion)? Finally, how to successfully transfer knowledge from training a model on one modality to another (co-learning)? Many data fusion techniques, such as the Kalman filter, Bayesian inference, and early fusion, can be found in the literature. Unfortunately, each solution is somewhat limited to low-dimensional or homogeneous data. Therefore, the proposed comparison does not cover them and focuses on the most flexible model-independent methods that can work with any data type.

The following chapter investigates state-of-the-art data fusion methods based on artificial neural networks in robotics-oriented tasks. In the experiments, multiple scenarios were employed, like grasp outcome classification, haptic and visual fusion, haptic-only signals fusion, and multiple scenarios where the input modalities were subjected to noises.

4.2 Related work

4.2.1 Data fusion approaches

In the review [122], the authors separated the multi-modal fusion methods into subfields – model agnostic and model-based. In the following work, experiments focused on the model agnostic methods because, typically, they are more general and widespread among roboticists. In the experimental section, chosen deep learning methods consist of three major categories of data fusion – early (data-level), intermediate (feature-level), and late (decision-level) [131]. Additionally, at least two of them might also be combined into one hybrid fusion method [122]. In [132], the authors systematically divided different sensor fusion methods.

Early fusion Generally, this type of fusion is based on combining information from different modalities at the very beginning of processing, e.g., by concatenating input time series (if possible). This method enables the deep learning model to *understand* low-level interactions between modalities and capture them together to produce meaningful features for further layers. However, a significant limitation exists for such approaches, i.e., when input signals or low-level features from initial layers do not fit each other because they are heterogeneous. For example, it is not obvious how to incorporate 2D images with a 1D time series into one feature space. It would be possible to concatenate both modalities by flattening the 2D image into a 1D vector. However, it would cause a loss of information about each pixel's position and significantly decrease that modality's information gain. For that reason, an early fusion approach was not incorporated in the experimental section, as it generally does not apply to all data types in the comparison.

Feature-level fusion Combining different data representations at some high level of abstraction is a common practice in machine learning for data fusion.

This technique allows for combining heterogeneous data from different modalities and lets the predictive model process joint representation. Among others, that type of fusion is popular in robotics-related areas such as object recognition [133] and scene recognition [134]. In [135], authors proposed a method for multi-modal data fusion applied to motion planning. In [124], authors employed an early multi-modal fusion for the contact-rich manipulation task. Their compelling results were appreciated, and the authors received the best paper award at the International Conference on Robotics and Automation (2019). The authors of [136] proposed another method of feature-level fusion, where instead of concatenation, they fused only some random parts of feature vectors from different modalities. Alternative intermediate fusion approach [137, 138] use so-called Tensor Fusion Networks, which often appear in multi-modal sentiment analysis literature. However, to the best of the author's knowledge, they were not used in robotic applications yet. This type of deep learning model suffers from low computational efficiency; however, the follow-up work on the LMF model in [139] further addressed this problem. The authors exploited a tensor decomposition to reduce the number of model parameters.

Late fusion Generally, late fusion methods manage to work with any data type, similar to feature-level fusion methods. They do not combine information but only the outputs of separate models used to process different modalities. The work [140] presents an outstanding review of late fusion techniques. Authors of [141, 142, 143] showed a typical plan of a late fusion. In [142, 143], the authors presented the late fusion method to process RGB-D data in object detection and discovery. The work [141] showed the method to combine images and point clouds for the semantic segmentation of the urban environment for autonomous vehicles. In [144], the authors proposed a late fusion deep learning model, which took into account the influence of a data deterioration on the model's decision and used a noisy-or operation to integrate these decisions. The experimental section of the following work also investigates the robustness of fusion methods.

Hybrid fusion This is an alternative approach to data fusion to the ones previously presented. It integrates information from different modalities at two or more levels. Authors of [127, 128] presented methods of such approaches for robust semantic segmentation in the outdoor environment for autonomous navigation. The authors proposed a convoluted mixture of deep experts that uses importance weights determined by a specialized gating network. This module, in turn, uses a feature-level representation of all modalities. The Mixture of Experts (MoE) can choose which modality should influence final segmentation more based on these weights.

4.2.2 Fusion robustness

Multi-modal and robotics-related literature very rarely considers fusion robustness. However, the author of the following thesis finds this problem urgent to solve regarding deploying an autonomous robot to the real world. Only several papers [136, 144, 145, 146, 147, 148] took into account the non-nominal conditions of a multi-modal fusion and provided some analysis of fusion robustness to data corruption. Such degradation could occur due to sensor noise, failure, or unexpected conditions like e.g., electromagnetic influence of motors used in industrial areas. The method presented in [127] can potentially take into account mentioned data degradation and express its *beliefs* in terms of the weighted model's decisions. However, the authors of this work did not elaborate more on that matter. Although one can find literature on the robustness of multi-modal fusion methods in robotics, it is noticeable that this area does not have any comprehensive and collaborative benchmark of fusion principles.

4.3 Proposed solution

Typically, robots operating in the real world have at their disposal multiple onboard sensors like cameras, F/T measurement devices, LiDARs, IMUs, and others. They all play a significant role in robotic perception tasks, including haptic perception or scene reconstruction. However, they also produce a vital piece of information that needs to be efficiently processed to exhibit to ensure robustness against disturbances. As the volume and dimensionality of sensory-feedback increase, it might be troublesome to manually design a multi-modal data fusion system that can handle heterogeneous data. Recently, multi-modal machine learning became an emerging field with research focused mainly on analyzing vision and audio information. Although, from the robotics perspective, haptic sensations experienced from interaction with an environment are essential. The following experiments focused on four learning-based fusion methods and three datasets containing haptic signals, images, and robots' poses. Tests related to grasp outcome classification, texture recognition, and – most challenging – a multi-label haptic adjectives classification based on haptic and visual data. Conducted experiments were focused not only on the verification of the performance of each method but mainly on their robustness against data degradation. Such degradation of sensory feedback might occur when the robot interacts with its environment. Additionally, the data augmentation technique was validated to increase the robustness of data fusion methods.

4.4 Multi-modal fusion from the robotics perspective

4.4.1 Problem formulation

The following section investigated three types of recognition problems present in the literature.

Binary classification of a grasp outcome Let the $f_{binary}\colon \mathcal{S} \mapsto \mathcal{C}$ be a grasp outcome prediction function that assigns a Boolean value representing the predicted grasp outcome. It is based on sensory feedback from the set $\mathcal{S}\colon \{s \in \mathbb{R}^{3\times n}\}$, where $3 \times n$ represents axes of the input data stream created with three tactile sensors mounted on a gripper.

Multi-class classification of textures Let the $f_{single}\colon \mathcal{S} \mapsto \mathcal{C}$ be a grasp outcome prediction function that assigns a discrete value from the class counter-domain $\mathcal{C}$ to a sensory measurement from the set $\mathcal{S}\colon \{s \in \mathbb{R}^{1}\}$ of signals registered while the unconstrained motion of a haptic sensor on a variety of surfaces.

Multi-label classification of haptic adjectives Let $f_{multi}\colon \mathcal{S} \mapsto \mathcal{H}$ be a multi-label haptic classification function that assigns a subset of haptic adjectives from the set of strings $\mathcal{H} = \{h_1, h_2, ..., h_m\}$ to a signal sample from the sensory domain $\mathcal{S}\colon \{s \in \mathbb{R}^{2\times n}\}$, where $2 \times n$ is a number of axes of the sensory feedback from two distinct sensors. Haptic adjectives assigned to each sample are not mutually exclusive, and each example might have an undefined number of haptic adjectives ranging from 1 to m. In this task, the sensory domain $\mathcal{S}$ contains heterogeneous data composed of RGB images of household objects with associated 2×19-axes sensory feedback from the tactile sensor's electrodes.

Given the problem formulations above, the following components could be introduced:

- an implementation and thorough comparison of four different deep learning methods for the fusion of multi-dimensional signals, including non-heterogeneous data – Late, MoE, Mid and LMF, that approximate functions f_{binary}, f_{single}, and f_{multi} in respective tasks;
- research on the impact of the data deterioration and leading modalities on the predictions of each model in the aforementioned tasks;

4.4.2 Multi-modal datasets

BioTac Grasp Stability Dataset (BiGS) This is a dataset that consists of 2000 samples of sensory feedback from the BioTac [29] for the grasp-stability prediction. Haptic signals come from grasping three objects: a ball, a box, and a cylinder. Each was associated with the registered outcome – a *success* or a *failure*. The authors of the following dataset used three bio-inspired BioTac sensors mounted on the fingers of a gripper and a F/T sensor mounted on the wrist to gather tactile sensory feedback. The experimental section of the following work shows the results of grasp outcomes based on the gripper's positions, orientation, and registered force signals. Each motion had a constant length by using the Fourier method to crop them. Eventually, each signal consisted of multi-dimensional time series with a length equal to 1053 time steps. Positions were 3-axis vectors, while orientation was 4-axis quaternion readings, and force readings were composed of 3-axis series. The training dataset for the cross-validation included 3197 signals from each modality, while the test set 801 such samples. All samples were independent of each other.

Penn Haptic Texture Toolkit (HaTT) The toolkit [3] consists of 100 different textures photographed and presented as RGB images. Each photo had associated normal force, acceleration, and position signals registered while the unconstrained motion of an impedance-type haptic device SensAble Phantom Omni [37]. Tab. 4.1 presents classes chosen for the experiments.

Table 4.1: Textures from the HaTT chosen for the experiments.

ABS Plastic	Aluminum Foil	Aluminum Square	Artificial Grass	Athletic Shirt
Binder	Blanket	Book	Brick 1	Brick 2

Signals in the HaTT dataset register the motion of a haptic device's tool-tip on different surfaces for 10 seconds. The experimental section utilizes normal forces, acceleration, and velocity signals as input modalities. To combine 3-axis signals into a single axis, the authors of the dataset used the method DFT321 [149]. Hence, the experiments involve this flattened representation. The authors motivated the dimensionality reduction because humans cannot sense the direction of high-frequency vibrations [150]. The training dataset had 8000 signals, while the test set included 2000. All signals were fixed-length (200 time-steps each), while the number of samples inside each class remained balanced.

Penn Haptic Adjective Corpus 2 (PHAC-2) The experimental section regarding this dataset investigates the problem of multi-label classification of haptic adjectives. The authors of [151] proposed an initial version of the PHAC-2 dataset, which has been updated in [4]. The following dataset includes 53 objects photographed from 8 different views. Each photo received a corresponding haptic signal from squeezing an object with two BioTac sensors mounted on a gripper. Moreover, every example corresponds with several haptic adjectives used as labels. In the dataset, there were 24 haptic adjectives in total. Fig. 4.2 presents the histogram of classes of the balanced dataset.

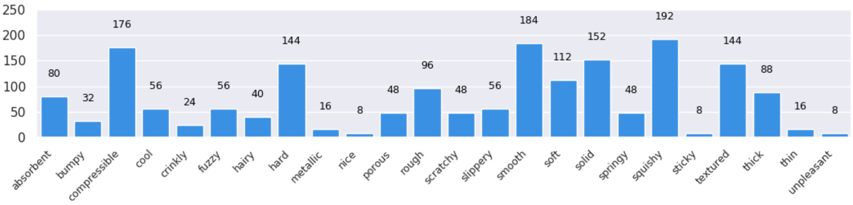


Figure 4.2: Occurrences of each adjective in the PHAC-2.

In the experimental section, one had to ensure that the adjectives distribution remained fixed among train and test subsets. The iterative stratification method [152] enabled us to avoid over/under-representing haptic adjectives in the train/test subsets. The such imbalance would lead to a significant decrease in the prediction performance and misleading results. Hence one can find this procedure of paramount importance. An RBG image with the spatial resolution of 224 × 224 with two raw signals from 19-electrode arrays from both BioTac sensors represented one input sample. All time series remained fixed-length with 67 values each. Eventually, 265 data samples were in the training dataset, while 159 appeared in the test set.

4.4.3 Fusion methods

The literature review shows several model-agnostic data fusion techniques, but the experimental analysis incorporates four of them, as it was shown in Fig. 4.3. Due to the simplicity and popularity of late and intermediate fusion, they inevitably appeared in the benchmark. The Mixture of Experts (MoE) [127] was another candidate for the experimental section. The proposed architecture was similar to the late fusion one, but it also can determine modality importance based on latent representations. The Low-Rank Multimodal Fusion (LMF) is a novelty in robotics, which appeared in the experimental analysis because of promising results in other areas, such as sentiment analysis. All of the chosen methods work on embeddings in the feature space. An encoder was a backbone for all models that transformed input signals into a 10-dimensional latent space. Each fusion method obtained N latent vectors $L_1, L_2, \ldots, L_N$ fed to selected network architectures. Methods to data fusion examined in the experimental validation were presented schematically in Fig. 4.3 and described in detail below.

Late Fusion Generally, that method's core operation principle is to process each input modality separately and combine predictions at the very end, assuming they are equally important. In [132], authors described this merging process at the decision level. According to Dasarathy's fusion classification [153], this approach is called the Decision In-Decision Out (DEI-DEO). Each latent vector was processed separately using neural networks in the experimental section. The overview shown in Fig. 4.3 illustrates these networks as arrows. This type of architecture was proposed to obtain predictions for each modality $p_1, p_2, \ldots, p_N$ in the form of logits. Next, these logits were summed up and transformed into class probabilities using a softmax function.

Mixture of Experts (MoE) This method works according to the same principle as the Late Fusion, but it can also determine input modalities' importance weights through the gating network. That judgment was encoded in a vector w representing weights, such that $\sum_{i=1}^{N} w_i = 1$. In contrast to the Late Fusion, corresponding weights influence predictions from modalities by multiplication. At the top, an MLP takes all latent vectors and produces final predictions w. That architecture potentially facilitated a neural network to learn reactions to the input data degradation by assigning lower weights to the degraded modalities. On the other hand, if the data deterioration did not occur during a training phase, there was a possibility that the MoE would put too much emphasis on the modality affected by some noise during testing, which might result in false predictions.

Intermediate Fusion (Mid) The fusion of information carried by individual modalities happened by concatenating their representations in the latent space. This shared representation was processed further to obtain a prediction based on common features. Authors of [132] presented a merging process at the feature level. According to Dasarathy's classification, this approach is the Feature In-Feature Out (FIFO). That method lets a fusion model draw information from all modalities in the latent space and process them freely. The Mid method would also be able to gain some robustness to the data degradation during the training, as it could learn to reduce the impact of degraded modalities. However, in contrast to MoE, its robustness and decisions were not interpretable at any stage of calculations.

Low-Rank Multimodal Fusion (LMF) This tensor-based approach for multi-modal fusion primarily focuses on revealing interactions between features extracted from different modalities. This technique aims to create some high-dimensional tensor-based representation by taking outer products over the uni-modal latent vectors $L_1, L_2, \ldots, L_N$. Then that representation linearly maps to the low-dimensional space using learned weights and biases. Fusion methods based on outer products typically suffer from computational inefficiency as tensor weights with the number of multiplications scale exponentially as the number of modalities. Nevertheless, the technique proposed by the authors of [139] does not use a high-dimensional weight tensor directly with the tensor representation of the data. The authors proposed decomposing tensor weights into N sets of modality-specific factors to improve the method's efficiency. Such decomposition significantly reduced the number of computations, as it lets to map from feature space to predictions directly. Moreover, it does not require any more explicitly creating any high-dimensional tensors.

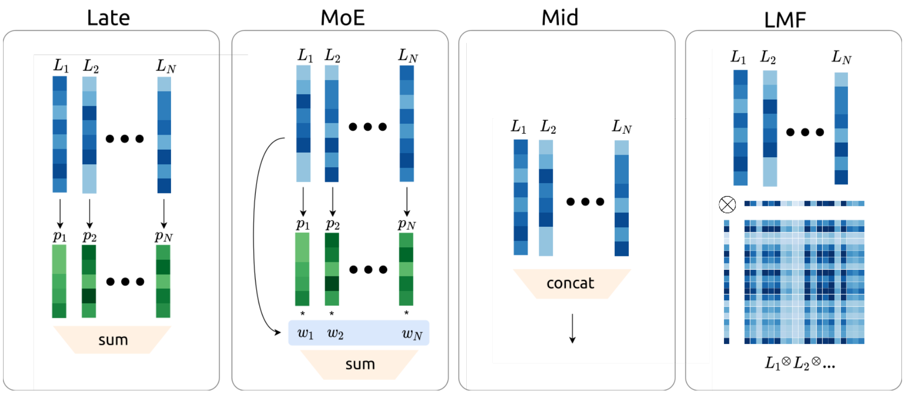


Figure 4.3: Multi-modal fusion architectures used in experiments. From left: Late Fusion (Late), Mixture of Experts (MoE), Intermediate Fusion (Mid), Low-Rank Multimodal Fusion (LMF). Arrows represent transformations with neural networks, while L_i, p_i, w_i denotes latent vectors, predictions, and trainable weight associated with i-th modality.

4.4.4 Discussion on chosen methods

The four fusion methods presented in the following section have their advantages and disadvantages. Late Fusion is a relatively simple-to-implement method that does not require complex architectures and allows each modality to be processed independently. However, it assumes that all modalities are equally important, which may not be true in many scenarios. On the other hand, MoE gains an additional degree of freedom when learning the importance of each modality and adjusting its predictions accordingly. Still, it may assign too much weight to noisy or irrelevant modalities, leading to incorrect predictions. Mid allows the fusion model to draw information from all modalities in the latent space, which can improve robustness to data degradation during training. However, similar to MoE, its decisions are not interpretable again. Finally, LMF aims to reveal interactions between features extracted from different modalities, but it suffers from computational inefficiency. The proposed decomposition in [139] significantly reduces the number of computations, but it still requires mapping from feature space to predictions directly. Thus, the choice of the fusion method should depend on the specific application's requirements and characteristics of the input modalities.

All models in the experimental section share a comparable number of learnable parameters. As datasets in the following experiments contained two heterogeneous types of data, i.e., time series and images, 1D convolutional layers (Conv1D) followed by the LSTM units processed time-series, while the FC at the top returned final predictions. Similarly, images were processed using 2D convolutional layers (Conv2D) with a few FC layers on top.

Similar neural network architectures were used for the BiGS and HaTT datasets to produce latent vectors – 3 × Conv1D layers with 64 filters of size 5 × 5 with stride equal to 2, followed by the LSTM layer with 32 units, and 2 FC layers with 128 and 10 neurons respectively. However, in the case of the BiGS dataset, for Mid, Late, and the MoE, the number of units in the last FC layer was changed to 2, as it was a binary classification. In the case of the MoE (for the BiGS and HaTT), the number of convolutional layers filters was reduced. This architecture uses an additional layer to produce w_i. This network for all datasets had the same architecture, namely, 3 × FC layers with 128, 64, and N units, where N is the number of modalities. Similarly, in the Mid fusion to process the concatenated latent vectors into the predictions, but in the last layer, the number of units was equal to the number of classes.

In the experiments regarding the PHAC-2 dataset containing time series and images, heterogeneous data samples had to be processed together. Models used for time series had similar architecture as for BiGS and HaTT datasets, however with 24 neurons in the last FC layer and 2 × Conv1D layers for all methods except the LMF. While processing images, the neural network had 2× Conv2D layers with 64 filters of size 5×5, stride 2, followed by 2 × FC layers with 128 and 24 neurons. Again, except the LMF, which had 10 neurons in the final layer.

The code used in the experimental section to create the following comparison is available online [1]. It contains additional implementation details of fusion methods and the architectures of neural networks used in the experiments.

[1] `https://bitbucket.org/m_bed/sense-switch/`

Part III

Experimental verification

Chapter 5

Material recognition

5.1 Supervised stiffness estimation

The main focus of this section is to introduce the results from the experiments on supervised stiffness estimation, which will be the subject of the subsequent section. I encourage readers to refer to the section 2.4 for formulating this topic.

5.1.1 Comparison of architectures

The experimental section contains the results of three types of neural networks tested using simulated signals to choose the best one for further experiments. Experiments on the performance of compared models focused on Mean Absolute Error (MAE) defined as:

$$MAE(y_{true}, y_{pred}) = \frac{1}{n} \sum_{i=1}^{n} |y_{true_i} - y_{pred_i}| \tag{5.1}$$

Where y_{true} represents the true labels, y_{pred} represents the predicted labels, n represents the number of samples in the dataset, and $|\cdot|$ denotes the absolute value. The second evaluation metric was Mean Absolute Percentage Error (MAPE) defined as:

$$MAPE(y_{true}, y_{pred}) = \frac{100\%}{n} \sum_{i=1}^{n} \left| \frac{y_{true_i} - y_{pred_i}}{y_{true_i}} \right| \tag{5.2}$$

Tab. 5.1 presents values of MAE and MAPE from the cross-validation. The objective of the comparison was to evaluate the performance of the proposed algorithms and choose the best-performing method for further experiments on shape generalization and to close the reality gap.

Table 5.1: Comparison of three architectures according to MAE/MAPE metrics.

	CNN		**CNN-LSTM**		**CNN-Bi-LSTM**	
k-fold	**MAE**	**MAPE**	**MAE**	**MAPE**	**MAE**	**MAPE**
I	19,1	2,4	6,2	0,8	6,2	0,8
II	11,8	1,6	5,4	0,7	5,4	0,7
III	15,1	2,2	7,8	1,1	7,8	1,1
IV	14,6	1,9	6,7	0,9	6,7	0,9
V	18,1	2,1	6,2	1,0	6,2	1,0
MEAN	15,7	2,0	6,8	0,9	**6,5**	**0,9**
SD	2,9	0,3	0,9	0,2	**0,7**	**0,1**

The experiments showed the performance of three types of neural networks in a stiffness parameter estimation based on inertial and sensory feedback. We were able to choose the best one for further analysis. Initially, experiments restricted all models to using only the simulation dataset without any real-world data samples. Tab. 5.1 shows results from the cross-validation procedure on the simulation dataset to verify the consistency of the dataset and to find the best-performing model. The mean results of the MAE/MAPE show the advantage of LSTM models in the stiffness estimation using raw inertial recordings. Firstly, the CNN-Bi-LSTM is more accurate in its predictions than CNN, resulting in MAE of $6,5\frac{N}{m}$ and MAPE of $0,9\%$, which means the improvement over $9,5\frac{N}{m}$ and $1,1\%$ achieved by the CNN. Secondly, the resilience of the learning process also improved, and reduced deviations of errors obtained between cross-validation folds prove that statement. For CNN a standard deviation of results is $2,9\frac{N}{m}$ MAE and $0,3\%$ MAPE, while the CNN-Bi-LSTM reduced these metrics to $0,9\frac{N}{m}$ and $0,2\%$ respectively. Although the results for both recurrent models appear on the same level, the CNN-Bi-LSTM exhibited a slightly better performance resulting in a smaller MAE. It means a lower absolute error on average. Finally, any further experiments utilized that architecture.

5.1.2 Shape generalization

The experimental section includes more experiments concerning the simulation-only datasets to verify the capability of the CNN-Bi-LSTM in the stiffness estimation. They started from the cross-validation for chosen model and reported the MAE/MAPE for three different datasets in Tab. 5.2. Each test dataset consists of sensor readings from squeezing episodes of only one object to report outcomes on the shape-dependent regression.

The generalization capability of the CNN-Bi-LSTM and verification of its performance on different types of objects need some additional experiments.

Table 5.2: Results from experiments on the shape-invariant estimation of the stiffness parameter using CNN-Bi-LSTM.

k - fold	**Dataset**					
	Ball		**Box**		**Cylinder**	
	MAE	MAPE	MAE	MAPE	MAE	MAPE
I	20,3	2,0	24,1	1,8	15,6	1,8
II	29,6	2,6	12,9	1,6	15,8	1,9
III	27,1	2,0	22,8	1,8	16,0	1,9
IV	21,8	2,1	17,7	16,6	18,4	1,9
V	19,3	2,0	24,4	1,5	20,8	1,9
MEAN	23,6	2,1	20,4	4,7	**17,3**	**1,9**
SD	4,5	0,3	5,0	6,7	**2,2**	**0,0**

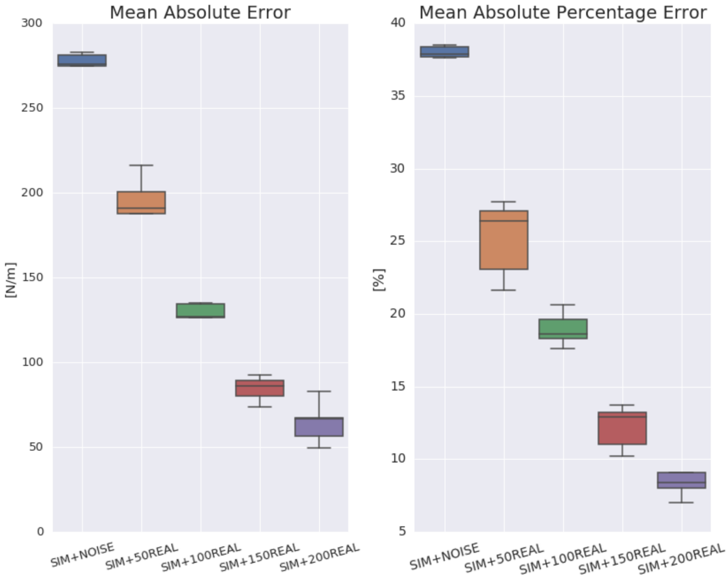


Figure 5.1: The box plot presents stiffness estimation performance metrics MAE/MAPE obtained on real-world test dataset. The number of real-world data samples included in the training dataset increases while the test error decreases. Boxes represent successive experiments and consist of the five-number summary of the result (from the bottom of each box): minimum, first quartile, median, third quartile, and maximum value.

Tab. 5.2 presents the MAE/MAPE from testing the network on three separate datasets, each of which includes only one shape of an object while training all at once. The results show that the model performed well in the shape-invariant stiffness parameter prediction and generalized to different shapes. The cylinder-shaped instances apparently let the Bi-LSTM achieve the lowest error than other shapes of $17,3\frac{N}{m}$ MAE and $1,9\%$ MAPE. However, box objects gave smaller values of MAE ($20,4\frac{N}{m}$) than ball-shaped objects ($23,6\frac{N}{m}$), while looking at the MAPE, the situation was the opposite, and some larger error was observed for boxes ($4,7\%$ / $2,1\%$). The model was inaccurate more often while predicting large stiffness values for boxes that resulted in the increased relative metric MAPE. For spherical shapes, the estimation quality decreased to small values that gave increased absolute measure (MAE).

5.1.3 Closing the reality gap

Table 5.3: The table presents performance metrics MAE/MAPE for best epochs from each of the cross-validation turns. Injecting even a small number of real-world sensor readings into the training resulted in a significant improvement in performance.

Experiment Name	**k - fold**										**MEAN**	
	I		**II**		**III**		**IV**		**V**		**MAE**	**MAPE**
	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE		
sim + noise	281,3	37,7	275,0	38,5	275,6	38,4	282,7	37,6	256,6	37,9	**274,2 ± 10,4**	**38,0 ± 0,4**
sim + 50 real	190,6	23,1	216,1	27,1	187,8	26,4	151,8	21,6	200,7	27,7	**189,4 ± 23,8**	**25,2 ± 2,7**
sim + 100 real	134,6	20,6	108,3	17,6	134,9	19,6	126,8	18,6	126,6	18,3	**126,2 ± 10,8**	**18,9 ± 1,2**
sim + 150 real	89,3	12,9	85,9	13,7	92,7	13,2	73,9	11,0	79,9	10,2	**84,3 ± 7,5**	**12,2 ± 1,5**
sim + 200 real	66,9	9,1	49,3	7,0	82,6	10,9	67,4	8,4	56,6	8,0	**64,6 ± 12,6**	**8,7 ± 1,5**

In the haptic recognition of physical parameters, data from the physics simulator seemed to resemble the real-world IMU readings only to some extent. Although the results from *sim + noise* tests were significantly worse than any of the *sim + real* trail, the mean MAPE 38% suggests that the correspondence between the simulation-only and real-world signals exists. Another important note is that MAE/MAPE values from each fold in the *sim + noise* trial remained near to each other. The model's result was similar for the entire dataset, as it was equally balanced in the stiffness parameters range. However, we cannot consider the reality gap problem to be solved yet, because the greatest improvement appeared in the experiments with the real-world sensor readings in the training dataset. In Fig. 5.1, one can observe the decreasing value of MAE/MAPE metrics as the number of real data samples are added to the training dataset. The experimental section does not include the results from the training on the real-world data only, as they would be incomparable with other experiments due to the low variability of the stiffness coefficient. Additionally, the number of data samples would be too low to assess a fair comparison in a real-world scenario. The lowest MAE/MAPE obtained in experiments on closing the reality gap were achieved for *sim + 200 real* trial and were equal to $64,6\frac{N}{m}$ and $8,7\%$. However, in the *sim + 50 real* experiment, the added number of real samples comprised only $1,2\%$ of the entire training dataset, but it gave the largest improvement among all experiments. The improvement was $84,8\frac{N}{m}$ and $12,8\%$ of the MAE/MAPE.

5.2 Unsupervised haptic recognition

The main focus of this section is to introduce the results from the experiments on unsupervised haptic recognition. I encourage readers to refer to the section 2.5 for formulating this topic.

5.2.1 Clustering of force measurements from the Touching dataset

The following experiments compare unsupervised learning methods in the clustering assignment of 3-axis force readings. These were gathered in equally distributed episodes of touching different materials lying on the hard surface. Fig. 5.2 shows the benchmark of different models tested on the Touching dataset. One can observe that DEC methods outperformed other, more classical approaches, showing almost 60% of clustering accuracy, 0.7 normalized mutual information score, and 0.65 purity. Only a slight improvement was observed for the DEC variant that clustered embeddings produced by the autoencoder. These were improvements achieved: 2.7%, 0.3%, and 3.1% accordingly.

Classical machine learning methods for the clustering task did not meet any requirements of real-world deployment, achieving clustering accuracy below 20%, while one of them - the Agglomerative Clustering method, was less than 10%. It means these methods are unsuitable for real-world time series from our experiments. Time dependencies, high signal noise typical for robotic applications, and a relatively small dataset prevent them from being useful for the clustering task.

However, one can observe that force measurements in the Touching dataset might not correctly describe touched material because they all lay on a hard surface. Hence, thin fabrics, like, e.g., linen bags, could take on the properties of the ground and become hard to recognize without supervision. On the other hand, hard materials, like fabric or plastic, might be difficult to differentiate based only on force measurements. That is why the Touching dataset was challenging. More experiments on unsupervised haptic recognition were needed

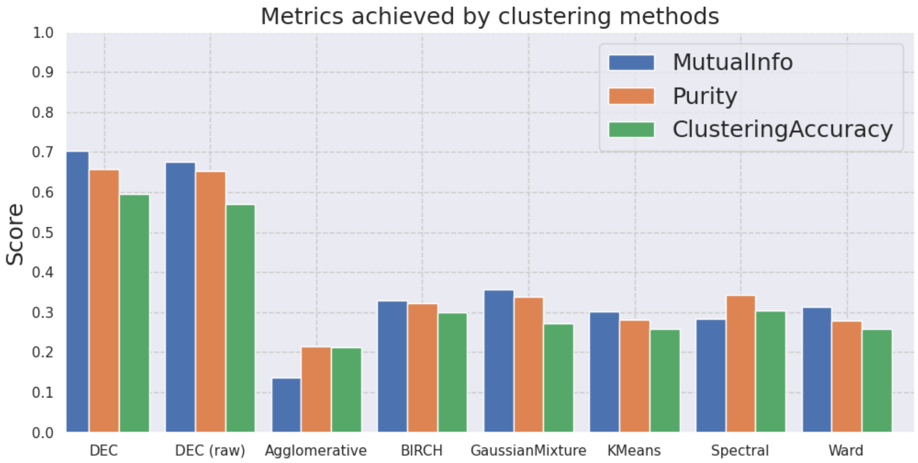


Figure 5.2: Evaluation metrics obtained for the clustering assignment of the Touching dataset.

using another dataset - preferably the one that isolates touched objects from the ground and ensures measurements from different directions to gain more insight into the object's physics.

5.2.2 Experiment design importance - BioTac dataset

More experiments were needed to verify the usefulness of unsupervised learning methods in haptic recognition. Fig. 5.3 shows the comparison of chosen methods for clustering haptic measurements from the BioTac sensors. Again, both DEC variants outperformed classical approaches by a large margin in all performance metrics, which proves that classical methods are not the proper choice for such a task.

However, all metrics of both DEC methods significantly improved compared to the experiments conducted on the Touching dataset. For the DEC clustering latent vectors from the autoencoder, the mutual information score achieved over 0.9, which means that the proposed clusters are statistically similar to the proper labels. That result is impressive, considering that there were 51 different classes in the dataset. Similarly, the purity score exceeding 0.8 shows that predicted clusters are homogeneous and mainly contain one object type each. Finally, the clustering accuracy of 79% shows that DEC clustering strategies of the unsupervised haptic recognition task are suitable for robotics applications demanding such a functionality. This accuracy score is calculated by finding the optimal assignment of predicted labels to clusters using the Hungarian algorithm and then computing the ratio of correctly assigned labels to the total number of samples. Unsupervised accuracy does not require the notion of classes as in supervised learning, and can achieve 100% when all samples from each true label fall into separate clusters.

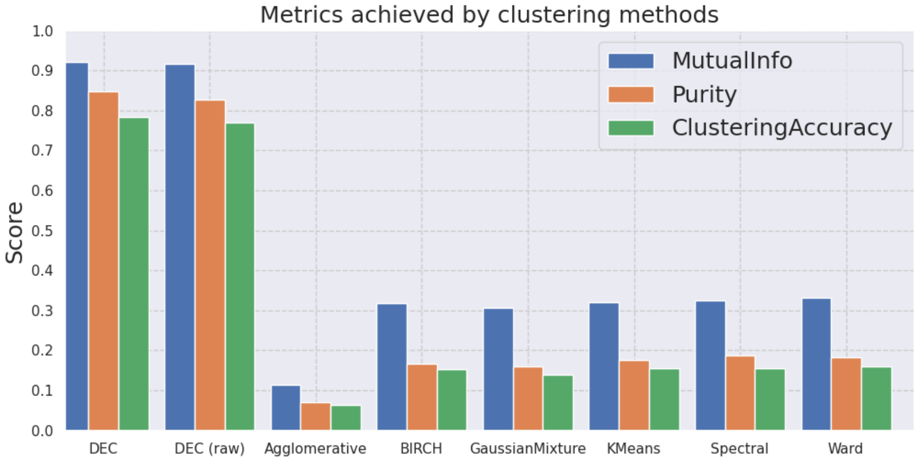


Figure 5.3: BiGS V2 consists of sensor measurements collected at a single time step, allowing DEC to operate on raw data without any encoding. This Fig. displays evaluation metrics for the benchmark methods in the clustering assignment.

Again, similarly to the experiments using the Touching dataset, DEC variant clustering latent vectors achieved slightly better results than the one working on raw measurements, exceeding its mutual information score, purity, and clustering accuracy by 0.5, 0.1, and 1.3%, respectively.

5.3 Discovering tactile dimensions - silhouette test

A human's sense of touch might carry complex information about the touched surface - not only about its hardness but also the type of texture, shape, or temperature. These factors might be categorized into *tactile dimensions* that comprise the psycho-physical perception of the world. In [52], authors conducted extensive experiments that yielded a definition of these categories: macro, fine roughness, warmness/coldness, hardness/softness, and friction. In the following experiments, the objective was to find the optimal number of clusters - if tactile dimensions exist, the optimal number of revealed sets would equal the number of dimensions. Thus, the experiment would prove or refute the thesis posed by the authors of [52]. Typically, we use silhouette analysis to find the optimal number of clusters in the clustering assignment. The technique involves calculating the silhouette coefficient (see 2.5.4) for each cluster sample, which measures how closely it matches its cluster compared to other groups. The silhouette coefficient ranges from -1 to 1, with a high value indicating that the example is well-matched to its cluster and a low value indicating that the sample is poorly matched and may be better suited to a different set.

Previous tests showed that only the DEC was suitable for the silhouette analysis on the BiGS V2 dataset. However, that method required different numbers of expected clusters before the training, so it started from scratch for every trial. These experiments involved DEC variant working on latent vectors. Tab. 5.4 presents silhouette scores obtained after consecutive training procedures. The highest score achieved model grouping data samples into three groups.

	Num. of clusters					
	II	III	IV	V	VI	VII
Silhouette score	0.467	**0.475**	0.406	0.301	0.300	0.264

Table 5.4: Silhouette score for the DEC for different numbers of clusters.

Fig. 5.4 visualizes embeddings using T-SNE and silhouette scores for every sample in the dataset. The model trained to group data into three clusters achieved the highest result. BioTac sensor is one of the most sophisticated haptic sensors on the market. However, compared to the human's fingertip, its 24 sensing electrodes might be too sparsely placed to recognize an object's macro and roughness properly, so the clustering strategy is based on other tactile dimensions – warmness, hardness, and friction. That result partially proves the existence of tactile dimensions.

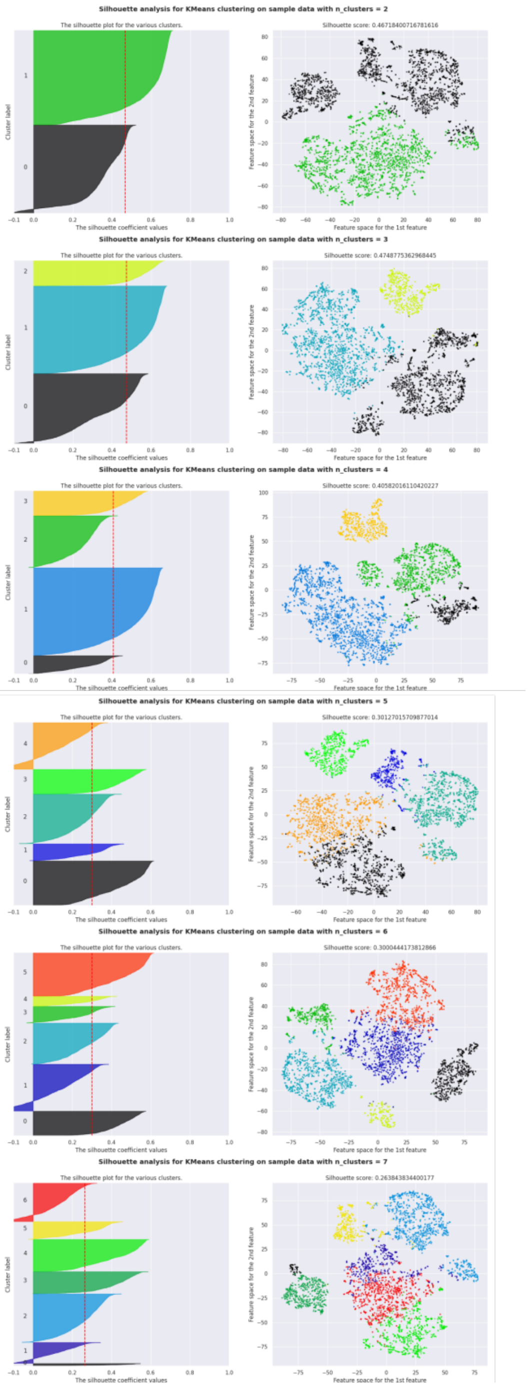


Figure 5.4: Silhouette test for every sample in the BiGS V2 dataset next to the visualization of embeddings using the T-SNE method.

Chapter 6

Terrain classification

6.1 Terrain classification using Transformers

The main focus of this section is to introduce the results from the experiments on terrain classification for walking robots. I encourage readers to refer to the section 3.4 for formulating this topic.

6.1.1 Real-life requirements for the terrain classification

In the vast majority of the research articles, overall accuracy is the state-of-the-art metric describing the performance of the supervised terrain recognition algorithm. However, when the distribution of classes in the dataset is not uniform, this measure will be skewed in favor of over-represented categories.

In the real world, an autonomous robot has to deal with numerous terrain types that naturally occur with different frequencies and want to acquire a reliable performance in all the considered terrains. To verify that classification algorithms are robust enough to operate in the real world, the experimental section presents a criterion of minimal accuracy Acc_{min}:

$$Acc_{min} = \min_{i}(Acc_i), \tag{6.1}$$

where Acc_i is the accuracy for an i-th terrain type. The goal of the Acc_{min} measure is to capture the classifier's performance on the most challenging terrain type.

The accuracy metric represents the system's ability to recognize the terrain but does not capture if it is viable to be deployed and used on a real legged robot. This chapter argues that the classification method must have a low computational burden and fast inference time for the robot to adapt its gait in real-time and prevent any possible damage. As each walking robot is different, the overriding assumption is that an inference time below 10 ms would satisfy these needs and be estimated based on the typical frequency of the control loops of the legs reaching hundreds of times per second. The neural network size should be as small as possible as a genuine autonomous robot needs GPU capabilities for a range of other tasks, e.g., object detection or segmentation. Apart from the general accuracy, we should also consider the accuracy of the most challenging terrain, model size, and inference time to determine the best classification method for the terrain classification task.

6.1.2 Accuracy evaluation on the Poznan University of Technology ANYmal dataset

In this section, to compare the HAPTR with the public results, the PUTAny dataset was split into train/test subsets. The test data came from a different run than the samples used for training. Tab. 6.1 presents the results obtained by the classical algorithms (KNN-DTW, ROCKET), classical deep learning solution (TCN), transformer-based approaches (HAPTR, HAPTR2), and the state-of-the-art CNN-RNN.

Table 6.1: (© 2022 ELSEVIER) The accuracy comparison measured on the test set of the PUTAny dataset.

Method	Variant	*Acc* [%]	Acc_{min} [%]
KNN-DTW	-	74.0	54.4 (PVC)
ROCKET	-	84.9	47.3 (PVC)
TCN	*Light*	84.5	68.7 (Art. grass)
	Base	86.9	65.1 (Art. grass)
	Large	87.5	72.3 (Art. grass)
HAPTR	*Light*	83.3	56.6 (Art. grass)
	Base	90.3	80.7 (Art. grass)
	Large	91.7	74.7 (Art. grass)
HAPTR2	*Light*	91.7	80.7 (Art. grass)
	Base	92.2	81.9 (Art. grass)
	Large	92.7	81.9 (Art. grass)
CNN-RNN [18]		93.0	86.7 (Art. grass)

The results achieved on the test set indicate that deep learning methods (TCN, HAPTR, HAPTR2, CNN-RNN) generally achieve better accuracy than any of the classical approaches (KNN-DTW, ROCKET). The most fundamental and universal KNN-DTW method performed the worst – 74%, while the ROCKET classifier improved its result to 84.9%. The overall satisfactory accuracy of the ROCKET came with a poor ability to perform on the most challenging terrain (PVC) of 47.3, which might be inadequate to complete the necessary gait adaptation. Among the deep learning methods, the different variants of the HAPTR outperformed the classical approach (TCN). Similarly, the improved HAPTR2 exceeded the result of the HAPTR. For all of these methods, we reported that increasing a network's size improved its accuracy and also improved Acc_{min}. Nevertheless, neither of the presented solutions could overcome the state-of-the-art CNN-RNN solution that reported the best accuracy of 93.0% with the best $Acc_{min} = 86.7\%$.

This comparison shows the model's accuracy without cross-validation (CV) as opposed to comparable works of [18] and [56]. The following experiments required an independent test set to truly measure the performance of a network in a setup resembling a real-world operation. Reporting results from cross-validation can still provide valuable information, such as the variance of the performance estimates. However, it may give a less accurate estimate of the model's performance on new data. The accuracy comparison comes from an experiment using n-fold CV, and the test accuracy originates from using an independent testing sequence to verify this claim. All analyzed methods achieved accuracy lower by several percentage points using testing sequence than n-fold CV. Most notably, even the best-performing CNN-RNN reported 94.1% when using CV [18] while we were only able to obtain 93.0% on the independent testing sequence.

6.1.3 Accuracy depending on the model size

Typically, the most extensive predictive models achieve the best performance on the chosen dataset. They are often too large to be trained or used in practical scenarios, thus being considered a dedicated solution for a particular dataset. Therefore, the applied deep learning community is currently more interested in efficiently formulating artificial neural networks and proving that these architectures are more capable than their predecessors with the same number of parameters in the case of EfficientNetV2 [154]. Fig. 6.1 presents a similar analysis with the accuracy presented in a function of the number of learnable parameters. The KNN-DTW was omitted in this analysis as it is a non-learnable method.

In Fig. 6.1, one can notice that the CNN-RNN achieved the highest accuracy, but it came with a price of its substantial number of learnable parameters. This method would not fit the restricted computational resources of a mobile robot setup. Still, it appears to be more efficient than the classical TCN approach

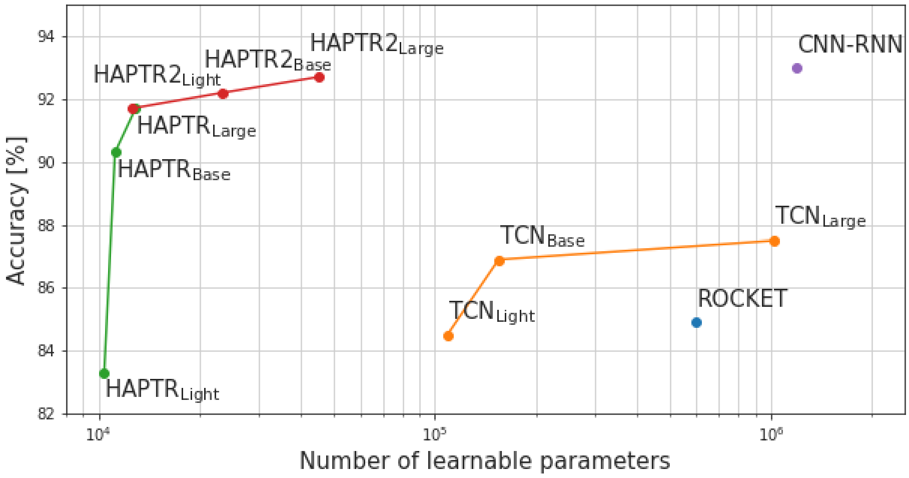


Figure 6.1: (© 2022 ELSEVIER) The accuracy as a function of a number of parameters reveals the efficiency of the applied method. Notice that the axis of parameters is in the logarithmic scale. In this context, HAPTR2 significantly outperforms other algorithms.

or even less efficient ROCKET. Both transformer-based solutions, HAPTR and HAPTR2, require an order of magnitude fewer parameters to achieve performance similar to CNN-RNN. The HAPTR2 contains more learnable parameters than the HAPTR, as it contains an additional attention mechanism but achieves significantly better results than the HAPTR. The introduction of such a module improved an internal representation of the input signal, which enabled better classification. Therefore, it is reasonable to use the family of attention-based solutions together with convolutional layers whenever the efficiency of the methods is the primary factor. Typically, convolution as a mathematical operation preserves high-frequency features, such as peaks in time series. However, the softmax function inside attention modules flattens the resulting representation and behaves more like a low-frequency filter [119]. Its output becomes complementary with a convolution operation. The combination of both layers together yielded better robustness and classification accuracy.

6.1.4 Inference time evaluation

The number of learnable parameters determines the ability to deploy networks in resource-constrained environments. Still, it might be challenging to compare networks properly as it depends on the chosen hardware setup. The network's size also impacts the training time, which we might consider irrelevant, as we trained them offline. The remaining aspect of a chosen model size is inference time. An inference time determines the real-world viability of the proposed solution, as having a prolonged processing time might be unacceptable from the perspective of the control that has been done based on this result. The Tab. 6.2 presents the processing times on CPU (Intel i7-9750H @ 2.600GHz) and GPU (NVIDIA GeForce GTX 1660 Ti Mobile) for all of the considered methods.

Table 6.2: (© 2022 ELSEVIER) Inference time for a single sample for evaluated terrain classification algorithms.

Method	Variant	CPU [ms]	GPU [ms]
KNN-DTW		1619.86 ± 11.11	–
ROCKET		180.62 ± 13.05	–
TCN	*Light*	1.37 ± 0.29	1.86 ± 0.05
	Base	2.97 ± 0.24	3.37 ± 0.04
	Large	16.82 ± 0.91	6.43 ± 0.06
HAPTR	*Light*	1.43 ± 0.09	1.61 ± 0.15
	Base	2.91 ± 0.22	2.73 ± 0.06
	Large	5.60 ± 0.59	4.87 ± 0.15
HAPTR2	*Light*	1.38 ± 0.03	1.53 ± 0.11
	Base	2.25 ± 0.05	2.13 ± 0.05
	Large	3.99 ± 0.09	3.39 ± 0.05
CNN-RNN [18]		11.68 ± 0.83	30.60 ± 5.30

Initially, let us consider a mobile robot equipped with a powerful CPU but no GPU. Fig. 6.2 presents results of the accuracy comparison as a function of mean inference times. In such a case, the KNN-DTW method performed poorly, and it took over 1600 ms to reach the desired classification result. This inference time was obtained for a selected size of training samples but would even increase if we added more signals, making this approach slow and unscalable for any real-world deployment. Nevertheless, the other non-deep learning ROCKET showed results in 180 ms, which is too long to implement any reaction-based behavior based on the obtained results. Deep-learning-based methods exhibited significantly lower inference times. The HAPTR2 reported the shortest CPU inference time. But it is also worth noting that the HAPTR in all variants and the TCN in *Light* and *Base* variants meet the previously stated requirement of 10 ms terrain classification time. In this comparison, the state-of-the-art CNN-RNN solution did not meet our criteria, exceeding the set threshold for the inference time, thus making it not suitable for a real-time operation.

Tab. 6.3 shows the results for a robot equipped with a CPU, and the trade-off between accuracy and inference time on a GPU. Classical methods (KNN-DTW and ROCKET) did not appear in the plot, as their implementations were not GPU-friendly. One can observe similar trends for deep-learning solutions to the processing performed on a CPU. Surprisingly, no acceleration was observed in

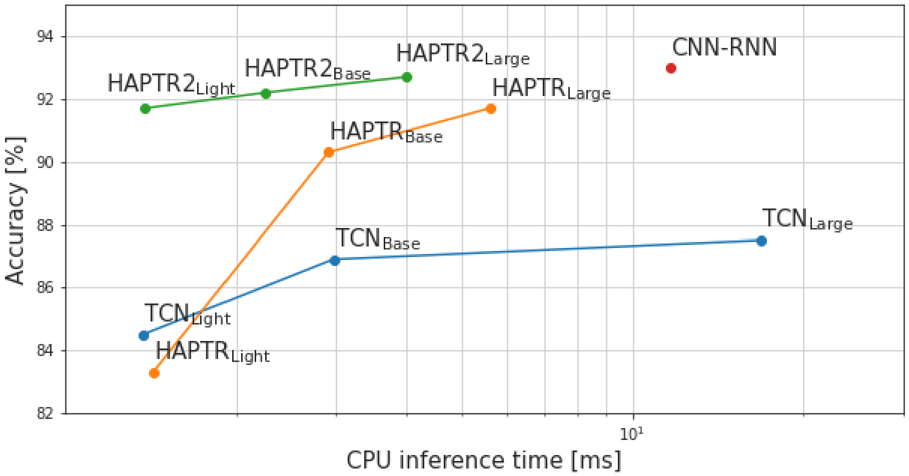


Figure 6.2: (© 2022 ELSEVIER) The accuracy of deep learning models as a function of a mean inference time on CPU. Notice that the axis of an inference time is in a logarithmic scale.

most cases when using a GPU. In the experiments, we defined the inference time as the total time needed for calculations and related tasks. It includes the time of transmitting data between the main memory and GPU's memory. This overhead is present in all the measurements of inference time included in the comparison. The reduced time is visible only for the TCN in *Large* that decreased below the accepted threshold. Much to the surprise, the inference time of the CNN-RNN increased on the GPU. The possible explanation would be insufficient optimization of the RNN model on this particular GPU architecture. Another probable reason would be sequential processing in recurrent units, which cannot fully utilize a GPU computational power – GPUs were designed primarily for parallel processing. Eventually, the HAPTR2 appeared to be the most efficient method based on the accuracy and the inference time on a CPU or GPU.

6.1.5 The choice and analysis of the best solution

Based on the results presented in the previous subsections, it becomes evident that the choice of the terrain classification method should not be based solely on its accuracy result. The decision on this topic should also be dictated by factors important to the real-world deployment, such as minimal classification accuracy on the most challenging terrain, model size, and its inference time on the target hardware. Overall, the chapter indicates that the HAPTR2 family provides the best efficiency and is regarded as the best choice for the trade-off

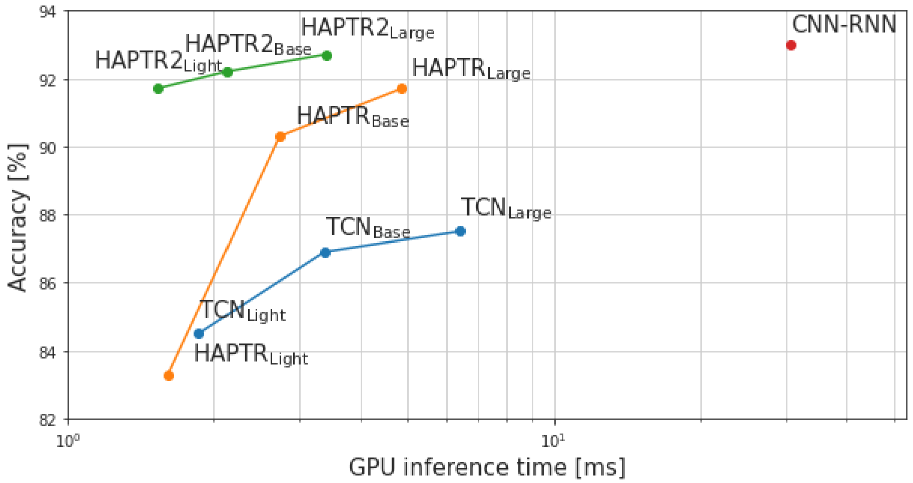
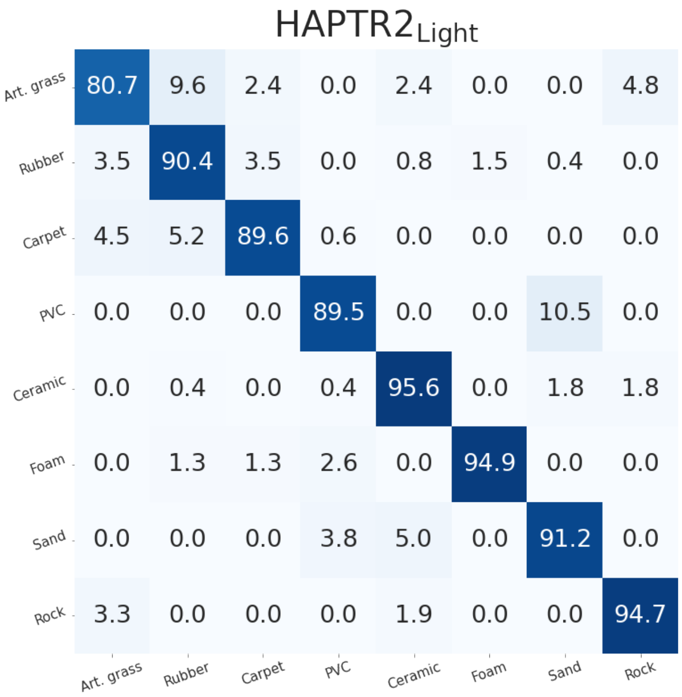


Figure 6.3: (© 2022 ELSEVIER) The accuracy as a function of a mean inference time on GPU. The axis of time is in the logarithmic scale.

between the accuracy and remaining requirements. Therefore, for further analysis, HAPTR2$_{\text{Light}}$ was chosen. Consequently, Fig. 6.4 presents the confusion matrix of this approach to provide more insight into observed performance.

The class that caused the lowest Acc_{min} for all tested deep learning models was artificial grass. One can observe that it was most often misinterpreted as the rubber – in 9.6% of predictions. Moreover, rubber was the most common mistake for a carpet, as the system was wrong in 5.2% of cases. One can assume that it is due to the designed terrain – a carpet sample was put on one slope, while squares of artificial grass lay close to the slope's beginning and end while rubber was on the second slope. That confirms that terrain recognition is more challenging in non-flat areas. There is still room for further improvements, i.e., data augmentation targeting these cases or utilization of orientation sensors to incorporate information about the inclination to predictions. However, the highest classification error among all classes was made for the PVC terrain incorrectly recognized as sand in 10.5% of predictions of that class. The root cause was that both of them were neighboring each other. Hence the sand particles were present in the PVC terrain, which misled the perception system. One can observe that a confusion matrix is related to the similarity between terrain and their placement on the map, which might be an informative cue in a localization task [18].



	Art. grass	Rubber	Carpet	PVC	Ceramic	Foam	Sand	Rock
Art. grass	80.7	9.6	2.4	0.0	2.4	0.0	0.0	4.8
Rubber	3.5	90.4	3.5	0.0	0.8	1.5	0.4	0.0
Carpet	4.5	5.2	89.6	0.6	0.0	0.0	0.0	0.0
PVC	0.0	0.0	0.0	89.5	0.0	0.0	10.5	0.0
Ceramic	0.0	0.4	0.0	0.4	95.6	0.0	1.8	1.8
Foam	0.0	1.3	1.3	2.6	0.0	94.9	0.0	0.0
Sand	0.0	0.0	0.0	3.8	5.0	0.0	91.2	0.0
Rock	3.3	0.0	0.0	0.0	1.9	0.0	0.0	94.7

Figure 6.4: (© 2022 ELSEVIER) The confusion matrix presents the per-class accuracy obtained with the $\text{HAPTR2}_{\text{Light}}$.

6.1.6 Robust robotic perception

The robot operating in the real-world environment must adapt to the changing conditions that cannot be predicted and thus trained before the deployment. To achieve the desired generalization ability and robustness, the HAPTR2$_{\text{Light}}$ model included the novel MAL. The MAL is responsible for dynamic adaptation of the weights of sensing modalities. This section contains the best-performing model's evaluation with and without this module to show its influence on predictions. Notably, not all modalities are equivalent. HAPTR2$_{\text{Light}}$ trained exclusively on torques achieved 88% accuracy in the test, while trained on forces only resulted in 60.0%. This experiment showed that torques are *leading modality* in the PUTany dataset. Nevertheless, experiments utilizing both modalities achieved the highest performance.

Table 6.3: (© 2022 ELSEVIER) Classification accuracy of the HAPTR2$_{\text{Light}}$ models trained with and without the additional modality attention module.

HAPTR2$_{\text{Light}}$	*Acc* [%]	Acc_{min} [%]
with MAL	91.7	80.7 (Art. grass)
without MAL	91.3	76.0 (Art. grass)

Firstly, as the MAL can work as a standalone module, its influence was verified on the overall accuracy performance. Tab. 6.3 presents the results of the HAPTR2$_{\text{Light}}$ model obtained on the PUTAny test dataset with and without the MAL. One can observe a minor improvement in the general classification accuracy (0.4%) due to the input modality weighting. However, the Acc_{min} metric was higher by 4.7% when the architecture included the MAL. That indicates that the model is more suited when training data is unbalanced.

Legged robots operate in diverse environments that can influence their sensory measurements. In most cases, operation in a novel surrounding results in inferior performance to the samples familiar to the system. The goal is to design a system that can operate despite these changes. The experiments evaluated the performance of the HAPTR in two simulated scenarios. Initially, the robot's payload was modified, changing the forces and torque distribution. Then, a sensory failure was simulated that might occur when a mobile robot traverses harsh terrain. We mimicked these cases by adding a particular type of noise to an already normalized measurement input determined to have zero mean and a unit standard deviation. The real-world effects occurring when the robot inspects the mine tunnels equipped with conveyor belts explain using a uniform noise in this experiment. The robot experiences noise from electrical equipment and motors, which were the root of the noise on the analogue sensors.

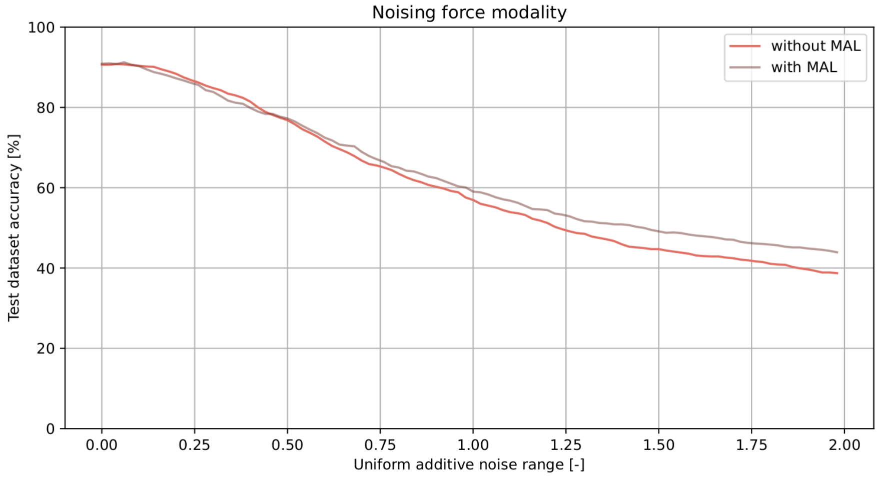


Figure 6.5: (© 2022 ELSEVIER) The simulation of the payload changes in the PUTAny test dataset and its influence on the performance of HAPTR$_{\text{Light}}$.

The payload change scenario simulates an additional increase in mass by adding a bias to forces. Each step consists of a stance and a flight. An assumption is that a robot's leg swings from 60 to 15 degrees from the normal to the ground during the stance to simulate the payload changes. In the following procedure, a simulated payload vector acts along the gravity vector and has an increased length, influencing the network predictions more. However, it is crucial to note that payload changes would also influence the torques. Regardless, the author of the following thesis did not find any effective method to simulate this effect. Hence torques stood unchanged. Firstly, the deep learning model fit to original data, but its weights stayed frozen during the simulation. Then, the simulated payload was added to the input force signal from 0.0 to 2.0. These values correspond to the robot's weight growing from an initial weight to three times the original weight. Such substantial change is unrealistic but demonstrates the generalization capabilities of the network. Fig. 6.5 presents accuracy depending on the simulated increase in mass. In this simulation, the HAPTR2$_{\text{Light}}$ with MAL achieved higher accuracy on the PUTAny dataset in a complete range of artificial payloads. Moreover, for an unexpected 3× change in weight, one can observe the drop in the accuracy by approximately 7.5% when MAL is not used.

The simulated case of a sensory failure introduced a uniformly distributed noise from a range of 0.0 to 0.25 to sensor measurements. Like the previous scenario, the experiment investigated performance on the PUTAny dataset measuring the accuracy at different noise levels. Fig. 6.6 shows the accuracy

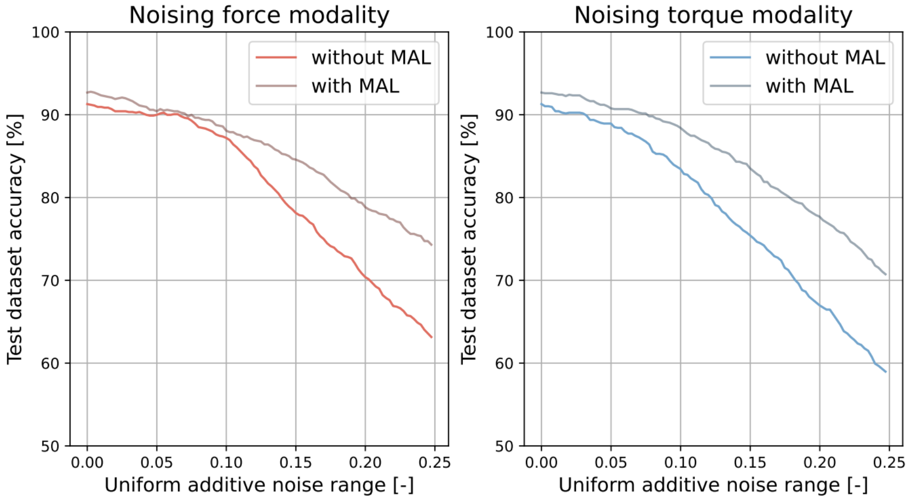


Figure 6.6: (© 2022 ELSEVIER) HAPTR2$_{\text{Light}}$ accuracy on the PUTAny test dataset, in which a uniform noise with an increasing range was added to one input modality (a force or a torque) to simulate a sensor failure.

obtained for the data degradation scenario. As one can observe, a significant improvement in the model's robustness can be noticed for both input modalities, achieving over 10% of accuracy improvement for the highest noise levels when it used the MAL. Moreover, the HAPTR was more robust to the changes in the force measurements proving that the torque measurements might have a higher impact on the final performance of the network.

6.1.7 Comparison to the state-of-the-art

The terrain classification algorithms for walking robots are mostly incomparable due to the different terrain types and hardware platforms. Nevertheless, it is changing due to the emergence of public datasets that facilitate impartial comparisons between methods. The results achieved were compared to the most recent method [56]. In that article, the authors evaluated RNNs+FC on the PUTAny dataset and their QCAT (made publicly available). The following experiments assessed the HAPTR2$_{\text{Light}}$ using the same evaluation procedure on both of these datasets using their cross-validation steps with the same data splits. Fig. 6.4 presents the obtained results.

The HAPTR2$_{\text{Light}}$ outperformed RNNs+FC on both datasets with an accuracy margin of 0.64% for PUTAny and 0.73% for QCAT datasets. The results showed similar standard deviations, thus proving that most methods would work similarly in the real-world environment. The HAPTR2$_{\text{Light}}$ is significantly

Table 6.4: (© 2022 ELSEVIER) Classification accuracy of the proposed HAPTR2$_{\text{Light}}$ and state-of-the-art RNNs+FC measured on the QCAT and PUTAny datasets with the 10-fold cross-validation providing mean, standard deviation (SD), and min and max values for each fold. The best results are bolded.

	Dataset	Mean [%]	SD [%]	Min [%]	Max [%]
HAPTR2$_{\text{Light}}$	PUTany	**93.85**	0.82	92.68	95.29
RNNs+FC [56]		93.20	0.89	92.06	95.39
HAPTR2$_{\text{Light}}$	QCAT	**97.33**	1.21	95.49	98.96
RNNs+FC [56]		96.60	0.89	95.49	98.61

smaller comparing the number of parameters and is more suitable for deploying on a real robot than RNNs+FC. Using the implementation shared by the authors of [56], the RNNs+FC consists of 395106 trainable variables with recurrent units. In contrast, the HAPTR2$_{\text{Light}}$ had only 12568 in total, which is over 30 times less. Moreover, an inference time was measured equal to 130.44 ms on a GPU and 38.44 ms on a CPU for the RNNs+FC. Similar to [18], the inference on a GPU took longer than on a CPU. GPUs are preferable processing units only when we process large batches of data. The experiments focused on the real-time robotics perspective, which prefers the inference of a single sample to reduce the delay between the measurement and the processed result.

Chapter 7

Robust multi-modal fusion

7.1 Multi-modal fusion from the robotics perspective

The primary objective of this section is to present the outcomes of the experiments conducted on robust multi-modal fusion with an emphasis on the robotics perspective of the problem. The following section will delve into this topic in more detail, and I urge readers to refer to 4.4 for a comprehensive discussion on the methodology and materials used.

7.1.1 Comparison of fusion methods

The first stage of experiments shows the performance of fusion methods on the BiGS dataset in the grasp outcome classification – a success or a failure. Input modalities include gripper positions, orientations, and 3-axis forces from a wrist-mounted F/T sensor. Tab. 7.1 presents the final results with mean accuracy [%] with its standard deviation among the consecutive folds. The best-performing models of Late (I-fold), MoE (III), Mid (II), and LMF (I) appeared in the assessment of their robustness against data degradation and influence of input data augmentation. The fact that the average results in subsequent folds were very similar means that differences in data distributions across folds were negligible.

Table 7.1: The classification accuracy comparison of four fusion methods performed on the BioTac Grasp Stability Dataset (BiGS) dataset.

	I	**II**	**III**	**IV**	**V**	**Mean**
Late	**88.9**	88.1	87.9	88.5	88.1	88.3 ± 0.4
MoE	89.0	88.3	**89.1**	87.6	88.4	88.5 ± 0.6
Mid	88.1	**89.9**	89.0	87.8	88.6	88.7 ± 0.8
LMF	89.6	88.4	88.0	88.4	**88.9**	88.7 ± 0.6

Cross-validation on the HaTT dataset was another experiment step. Results in the form of the classification accuracy [%] were reported in Tab. 7.2. In the following scenario, there appeared time-series—a squashed 1-dimensional representation of acceleration and velocity, together with a normal force acting on a haptic device's tool-tip. For the next experiments, the II-fold models were chosen for the Late, MoE, LMF methods, and the III-fold model for the Mid fusion approach.

Table 7.2: The classification accuracy comparison of four fusion methods performed on the HaTT dataset.

	I	**II**	**III**	**IV**	**V**	**Mean**
Late	79.5	**80.8**	79.4	78.3	79.5	79.5 ± 0.9
MoE	77.9	**78.9**	74.3	76.6	73.4	76.2 ± 2.3
Mid	78.9	75.4	**79.8**	78.3	76.6	77.8 ± 1.8
LMF	78.1	**80.9**	78.9	78.3	79.5	79.1 ± 1.1

The PHAC-2 dataset took part in the multi-label classification of haptic adjectives. Similarly, as in the [4], a performance metric was the Area Under Curve (AUC) – an area under the Receiver Operating Characteristic (ROC), which is a typical performance metric used in the literature about the multi-label classification. It measures how well the predictive model can distinguish between classes (haptic adjectives). Moreover, it considers correspondences between a sensitivity/specificity ratio to multiple values of a decision threshold. Generally, AUC-ROC is a metric that gives an overall evaluation of performance, taking into account all potential classification thresholds. A common interpretation of AUC-ROC is the likelihood of the model ranking a random positive instance higher than a random negative instance. In the AUC-ROC metric, a value of 1.0 refers to an excellent classification ability, 0 means that the model is always wrong, while 0.5 means that the model has no discrimination capacity. Tab. 7.3 presents the AUC-ROC metric achieved by fusion methods. Further experiments include the V-fold Late and MoE, I-fold Mid, and the IV-fold LMF model.

Table 7.3: The comparison of four fusion methods did on the PHAC-2 dataset. All values represent the AUC-ROC results.

	I	**II**	**III**	**IV**	**V**	**Mean**
Late	0.923	0.924	0.922	0.923	**0.925**	0.923 ± 0.001
MoE	0.923	0.919	0.919	0.923	**0.927**	0.922 ± 0.003
Mid	**0.929**	0.922	0.922	0.927	0.925	0.925 ± 0.003
LMF	0.896	0.898	0.902	**0.908**	0.900	0.901 ± 0.005

The important note is that the primary target of the experiments was to examine methods of homo and heterogeneous signals collection. Different sets of modalities were used from each dataset, and no modality was repeated between datasets even though, e.g., in the BiGS and PHAC-2, there were used the same BioTacs tactile sensors.

The mean accuracy of all methods tested on the BiGS dataset was around 88%, with insignificant differences between folds. Nevertheless, the most efficient fusion method in the grasp classification task was the LMF due to a decreased standard deviation among folds than the second method - the Mid. Tests on the HaTT exhibited a slight increase of a mean results variance among different approaches and standard deviations among folds compared to BiGS results. The best method for texture recognition was the Late fusion, which achieved a mean test accuracy of 79.5% between folds, additionally the least standard deviation equal to 0.9%. In the multi-label classification of haptic adjectives based on visual and haptic data, the Late, MoE, and the Mid fusion methods were extremely close to each other in terms of a mean AUC-ROC metric, achieving a result of 0.92. The LMF performance was marginally below the other.

The results from 5-fold cross-validation and tests on separate subsets on all tested datasets showed that data used in experiments were consistent, and there were no significant outliers among folds. That finding made it possible to perform reliable experiments and ensure a fair comparison. Mean values of metrics among different fusion methods may suggest that the type of data fusion affects the efficiency only to a limited extent. When all modalities are available and free of noise, it appears that more important was a reliable data preparation (e.g., ensuring a balanced distribution between classes in the train and test subsets, as well as between folds) for the training procedure than the fusion algorithm itself. The best-performing methods are marked with a bold font in Tab. 7.1, 7.2, and 7.3. Tested neural networks were trained end-to-end and functioned moderately well, exhibiting a great capacity to learn from large (BiGS, HaTT) and small (PHAC-2) datasets. Additional tests were conducted on the impact of each modality on predictions and verifying the robustness of each method against data deterioration.

7.1.2 Data degradation robustness

The research carried out in the following section brought conclusions on the capabilities of each fusion method to translate knowledge from one modality into another and revealed that often one leading modality exists. Each dataset has a modality of vital importance for the final results. To make this dependency visible, Fig. 7.1–7.3 present the results in the form of heat maps. Heat maps present changes in performance caused by the deterioration of the quality of one or more input modalities. In the following experiments, fusion methods were tested in scenarios described below, and each row in heat maps corresponds to one of the scenarios:

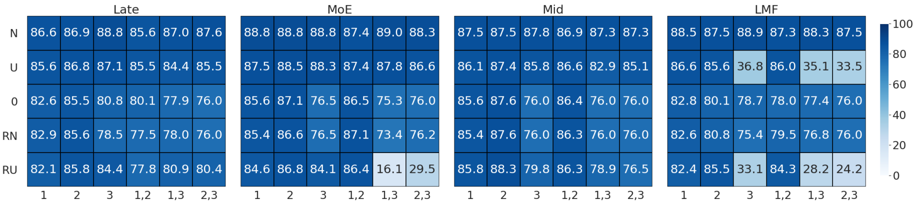
1. N—a Gaussian Noise N added to selected modalities with a 0 mean and 0.7 standard deviation;
2. U—a uniform noise U added to selected modalities that vary in the range (-0.5 to 0.5);
3. 0—setting zeros in place of selected modalities, what simulated a deactivated/broken sensor;
4. RN—replacing selected modalities with Gaussian noise N, with the same parameters as N from the previous scenario;
5. RU—replacing selected modalities with uniform noise U, with the same parameters as U from the previous scenario.

No fixed level of noise can be considered universally unacceptable for a normalized input signal, as it depends on various factors, such as the nature of the signal, the complexity of the classifier, the performance criteria, and the tolerance level of the application. Suppose we want to determine what it means for the noise level to be significant for the chosen classifier. In that case, it is necessary to perform experiments and analyze the classifier's performance under different noise levels. Because of that, the following experiments build upon a work of [95], where the authors conducted such experiments on the relevance of noise to the time series classification on the synthetic Cylinder-Bell-Funnel dataset [155]. The authors examined the effects of Gaussian noise added to the input signal. Their results showed that the first significant drop in the classification accuracy of their baseline method happened for the Gaussian noise, with a standard deviation increasing from 0.6 to 0.7, which can be considered significant. Thus in the following experiments, the level of 0.7 was also set. In addition, unlike Gaussian noise, due to the equal probability of values across the range for homogeneous noise, its range has been reduced relative to the standard deviation of Gaussian noise. Albeit, this change relies only on the author's estimates and, to the best of the author's knowledge, has yet to be considered in the literature.

Each heat map column was annotated by a number that specified affected modalities (e.g., by the added uniform noise). For each dataset, fusion methods were tested using three input modalities numbered as follows:

1. BiGS—*1*: gripper positions, *2*: gripper spatial orientations, *3*: 3-axis force;
2. HaTT—*1*: normal force, *2*: squashed acceleration, *3*: squashed velocity;
3. PHAC-2—*1*: images, *2*: raw electrodes from the 1st sensor, *3*: raw electrodes from the 2nd sensor.

Figure 7.1 shows heatmaps of accuracy results of selected fusion methods tested on the BiGS dataset. Heat maps enabled the inspection of knowledge alignment and translation of each method. In these tests, the best models from Tab. 7.1 were used.



Late

	1	2	3	1,2	1,3	2,3
N	86.6	86.9	88.8	85.6	87.0	87.6
U	85.6	86.8	87.1	85.5	84.4	85.5
0	82.6	85.5	80.8	80.1	77.9	76.0
RN	82.9	85.6	78.5	77.5	78.0	76.0
RU	82.1	85.8	84.4	77.8	80.9	80.4

MoE

	1	2	3	1,2	1,3	2,3
N	88.8	88.8	88.8	87.4	89.0	88.3
U	87.5	88.5	88.3	87.4	87.8	86.6
0	85.6	87.1	76.5	86.5	75.3	76.0
RN	85.4	86.6	76.5	87.1	73.4	76.2
RU	84.6	86.8	84.1	86.4	16.1	29.5

Mid

	1	2	3	1,2	1,3	2,3
N	87.5	87.5	87.8	86.9	87.3	87.3
U	86.1	87.4	85.8	86.6	82.9	85.1
0	85.6	87.6	76.0	86.4	76.0	76.0
RN	85.4	87.6	76.0	86.3	76.0	76.0
RU	85.8	88.3	79.8	86.3	78.9	76.5

LMF

	1	2	3	1,2	1,3	2,3
N	88.5	87.5	88.9	87.3	88.3	87.5
U	86.6	85.6	36.8	86.0	35.1	33.5
0	82.8	80.1	78.7	78.0	77.4	76.0
RN	82.6	80.8	75.4	79.5	76.8	76.0
RU	82.4	85.5	33.1	84.3	28.2	24.2

Figure 7.1: Heat maps present results obtained by chosen models from the first stage of experiments on degraded data from the BiGS dataset. Classification accuracy is in [%]. Each row corresponds to a different data degradation scenario. Columns are annotated by the indexes of affected modalities.

In Fig. 7.2, heat maps are generated for tests on the HaTT dataset. The influence of each modality on the final prediction is visible, and not every method can manage data degradation. Moreover, acceleration (2nd modality) plays a leading role. It resulted in a significant deterioration of classification accuracy when noisy or faded. On the other hand, removing other modalities from the input data stream did not affect the final accuracy.

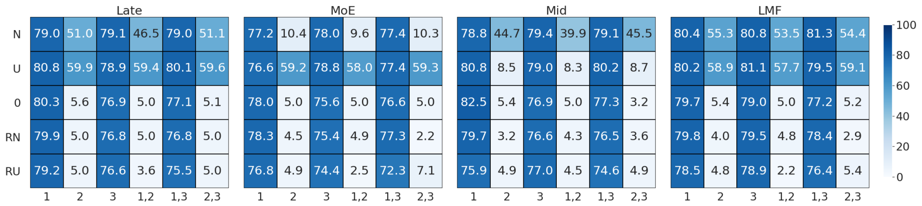


Figure 7.2: Heat maps include the accuracy [%] of a texture classification achieved while testing different fusion methods on degraded data from the HaTT dataset. Leading modality played a dominant role, which resulted in a decreased quality in case of its degradation.

Fig. 7.3 reports the AUC-ROC metric for the multi-label classification of haptic adjectives. Similarly, as in the experiments on the HaTT dataset, the leading modality is also visible. However, its correlations with other modalities played an even more dominant role in the final performance of the methods. By inspecting heat maps, one can observe that the most meaningful correlations for predictions are between images (1st) and raw electrode signals (second and third). On the other hand, noised interactions between both electrodes' time series only slightly influenced the classification accuracy.

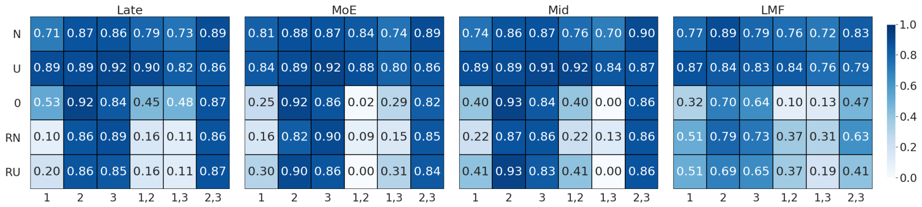


Figure 7.3: The AUC-ROC was reported for the multi-label classification task using the PHAC-2 dataset. The leading modality is visible, but the correlations between modalities also affect predictions.

All experiments on robustness revealed the existence of a leading modality, which means that one modality always played a dominant role in the discrimination between classes. Fig. 7.1 presents that for the BiGS dataset, MoE and LMF fusion methods exhibited significantly decreased performance when modality no. 3, which was a 3-axis force signal in combination with other modalities, was replaced by the uniform noise, and this is a leading modality in the BiGS dataset. The LMF and MoE were also sensitive to the uniform noise added that affected the force signal. However, the described phenomenon did not occur for other fusion methods – the Late and Mid. They appeared relatively robust for data degradation scenarios, exhibiting no more than 10% of a drop in the accuracy when the leading modality was noised, zeroed, or replaced by noise. In the case of MoE, the class discrimination dropped to 16.1% and 29.5%. It happens because the gating network increased the importance of a leading modality for prediction. After all, it played a dominant role in providing data.

Replacing the signal with other modalities with the noise emphasized the correlation of these signals, which caused prediction errors. However, MoE exhibited robustness for data degradation scenarios. It appeared to be sensitive mainly to the correlations between the force signal and two other modalities – hand positions and orientations. Similarities between modalities in the MoE were essential for the grasp outcome classification, not the leading modality itself. The LMF method achieved close results while testing. When the uniform noise replaced the force signal, the method was more prone to mistakes, and the interactions between modalities appeared meaningless. Additionally, the LMF was sensitive to scenarios that replaced the leading modality with noise and its combinations with other modalities and the noise added to the signals. This could be observed in the LMF heat-map in Fig. 7.1 looking at the results (36.8%, 35.1%, and 33.5%) in the U-row. The described phenomenon occurs because LMF is a tensor-based method that highly relies on outer products between unimodal representations inside networks. Thus the highest emphasis was put on inter-modality interactions. When finding these interactions were difficult/not possible, the LMF struggled to find a correct prediction for the grasp outcome evaluation task. One can observe that the type of noise introduced to the input data played a significant role in the prediction performance because the presented findings did not repeat for a Gaussian Noise. To explain that effect, one can speculate that during the training phase, in signals, there was already some noise present similar to the normal level, which resulted in higher robustness for such a data degradation. The achieved robustness was truly substantial, and it appears that a balance between the importance of modalities was paramount. Nevertheless, verifying that relevance is very challenging and involves many experiments.

Another dataset involved in experiments was the HaTT, and results gathered during that trial were reported in Fig. 7.2. Contrary to the BiGS dataset, one can observe that the second modality (an acceleration) caused a significant drop in the accuracy for all tested data degradation scenarios and fusion methods by inspecting heat maps. Hence, we consider acceleration to be a leading modality from the proposed set of input modalities. In the texture classification task based on haptic signals, all methods exhibited a similar performance and sensitivity to different disturbances. A lack of a legitimate acceleration signal (zeroing or replacing with a noise) always caused a decreased performance to the level of 5% for all tested methods, which suggests that in the proposed set of modalities, the domination of acceleration was tremendous. The rest of the signals did not provide meaningful information about the process under investigation. The late fusion and LMF heat-maps evaluation gave similar results for all data degradation scenarios. However, the MoE and Mid differ in terms of managing the added noise – MoE exhibited sensitivity to the appearance of the noise component in the leading modality. However, the Mid was fragile for the same phenomenon, but with the uniform noise added.

Fig. 7.3 shows the results of the multi-label classification of haptic adjectives performed on the PHAC-2 dataset. The biggest drop in the performance metric occurred for columns missing the first modality—an image (a leading modality). In the MoE, the fact that the lack of images was able to cause a total failure of the classifier achieving the result of 0 (which means that the modal was always wrong) again indicates that the gating network during training put too much emphasis on the dominant modality. Considering that every result below 0.5 level means that the classifier is often wrong than right on average. The Late, MoE, and Mid methods behaved similarly across all scenarios – the accuracy without a leading modality significantly decreased. The LMF functioned slightly differently, achieving relatively good results when a noise replaced the imaging modality, which is visible in the first column of the LMF heat map. However, it performed worst when other modalities were replaced by a noise/zeroed. Although it should be noted that it does not achieve 0 AUC-ROC as it happened in the case of MoE and Mid. Additionally, sometimes one can observe an improvement in the classification performance achieved when one modality was noised/faded. This phenomenon was reported, e.g., for the Mid when the second modality (raw electrode signal) was zeroed or replaced by a uniform noise. Comparing to Tab. 7.3 the improvement was 3%.

7.1.3 Data augmentation vs. leading modality

Generated heat maps from the previous stage of experiments revealed that every dataset contains one leading modality that had the biggest impact on the prediction. In the following experiments, 33% of randomly selected training samples of leading modalities in the training dataset were noised or faded. Half of the augmented examples were faded, and the other half noised with the 0 mean Gaussian Noise with a standard deviation of 0.7. This value should be related to the fact that the standard deviation of the data confines to 1 through data standardization. Again, The best models from previous experiments were re-trained on the same folds. Tab. 7.4 – 7.6 presents performance of methods used in the following experiments.

Firstly, fusion models were re-trained on the augmented training dataset and tested on the same versions of the test datasets. This experiment inspected the impact of the data degradation on the performance of test sets without it. An accuracy [%] and AUC-ROC values [-] were presented in Tab. 7.4.

Table 7.4: Outcomes of multi-modal fusion methods obtained for models trained on datasets containing noised/zeroed inputs from the leading modality, but tested on the dataset without such samples.

	Late	**MoE**	**Mid**	**LMF**
BiGS [%]	88.26	88.89	88.64	89.39
HaTT [%]	78.15	75.8	75.1	76.95
PHAC-2 [-]	0.92	0.93	0.93	0.91

Secondly, they were evaluated on test datasets with the same proportion of noised samples of the leading modality without zeroed samples. Tab. 7.5 reports results obtained during that trial.

Table 7.5: Tab. shows results obtained for models trained on datasets containing noised/zeroed inputs from the leading modality and a noised leading modality channel during tests.

	Late	**MoE**	**Mid**	**LMF**
BiGS [%]	88.14	88.64	88.51	89.26
HaTT [%]	65.85	69.95	65.3	68.3
PHAC-2 [-]	0.88	0.86	0.88	0.84

Finally, the experiments assessed the influence of zeroed leading modalities on fusion methods in the last stage. Tab. 7.6 contains accuracy and AUC-ROC metric results gathered on the test dataset with zeroed leading modality.

Table 7.6: Results obtained for models trained on datasets containing noised/zeroed inputs from the leading modality and a zeroed leading modality during tests.

	Late	**MoE**	**Mid**	**LMF**
BiGS [%]	85.77	88.76	86.27	89.01
HaTT [%]	54.85	57.05	53.95	53.25
PHAC-2 [-]	0.23	0.24	0.21	0.61

The data augmentation procedure increased robustness on noised and missing modalities for the BiGS. All methods gave similar results as during the tests on the data without any degradation applied on a leading modality. The proposed data augmentation procedure is sufficient to ensure robustness on noised/missing samples for the proposed input modalities.

However, the above statement is not always true, which is visible in results obtained for the HaTT dataset, when the mean decrease of accuracy was from 6% to 13% when comparing Tab. 7.4 to Tab. 7.5 and from 18% to 24% between Tab. 7.4 and 7.6. The results proved the same conclusions as before – the leading modality in the HaTT dataset possessed so much information meaningful for the discrimination between textures, and other modalities played only a supporting role for that task. Nevertheless, data augmentation still brought a significant improvement in results compared to data degradation scenarios shown in Tab. 7.2. In both tested variants, the best-performing method was the MoE, which indicates that the gating network learned to refuse predictions based on a degraded leading modality.

In the multi-label classification task on the PHAC-2, the Late, Mid, and MoE methods failed to assign haptic adjectives when the vision was missing. However, the LMF was able to find intra- and inter-modality interactions that led to the surprisingly good result of 0.61 AUC-ROC. It indicates that the LMF was the only method capable of assigning correct haptic adjectives more often than making mistakes. In tests involving noise-only samples, methods achieved similar results, and the performance metric dropped only by 4–6%.

Part IV

Conclusions

Chapter 8

Material recognition

8.1 Stiffness estimation using inertial sensors

Soft robotics solutions often take inspiration from nature, where animals exploit their flexibility to adapt to the environment, i.e., a soft-bodied octopus utilizes its tentacles to grab various objects that might have complex shapes. Such an approach's key feature incorporates embodied intelligence to execute a task. However, such a method leads to troublesome control strategies and difficulties in haptic perception because there is no feedback from the soft machine, as the idea of soft robotics primarily concerns mechanical design.

The following dissertation proposes to use contact feedback from a soft-gripper in the stiffness estimation task using IMU. Firstly, such an approach captures a typical inertial response of soft fingers from grasping objects of different shapes and physical parameters. Secondly, the object's physical parameters using data from IMU sensors is possible and beneficial due to the low cost of setup and no further need for sophisticated equipment. The deep learning solution solves the problem of grasped object stiffness estimation in soft robotics, introducing a novel approach that associates embodied and artificial intelligence. Their combination leads to a system robust to unforeseen and changing external conditions. While currently used methods of stiffness assessment exploit techniques of measurement or direct estimation, the method proposed in the dissertation characterizes the discovery of knowledge and causal relationships related to the characteristics of a given object and its physical features. Research on discovering knowledge acquired by neural networks may result in diagnosing the intuition behind humans' natural behavior in manipulating objects. It is likely that similar solutions, based on low-cost sensors and deep learning, may be successfully applied for robotic manipulation in everyday scenarios. In this work, there was published data and the implementation of neural networks used in the experiments. The author believes that it will inspire other researchers to delve into the research area of soft grippers and perception of the physical world based on tactile data in robotics.

8.2 Unsupervised haptic recognition

Although the DEC method outperformed other methods included in the benchmark, there is a need to mention the troubles when learning these models. Firstly, they express a high fragility on hyper-parameters, especially the batch size, which caused the learning process to collapse very early when parameter values were too high. A batch size of 256 prevents the model from learning, while a significant reduction to 16 leads to the presented results. The probable cause is the tendency of the clustering strategy to get stuck in the sharp minimum found very early in the training phase. The numerical proof for that was introduced in the following work [156]. The same applies to the learning rate, but the explanation is more straightforward and well-known - a too-large learning rate would cause the model to update its weights with a too-large step resulting in a deterioration of the training quality.

Moreover, experiments show that the Touching dataset was improperly designed for the considered task. As stated before, such a system's training procedure might be fragile and sensitive to the hyper-parameters choice. Any additional noise introduced in the input signals might destroy the learning curve. The situation in which different signals exhibit some false features, like, e.g., the high hardness of the soft object, because the sensor sensed the hard ground beneath them, might influence the training. That phenomenon caused poor results in these experiments.

The following benchmark also presented results from other clustering methods. The conclusion is equal for all of them - they do not apply to such a task. As haptic signals are transmitted through touch, they also might be ambiguous (different objects might feel the same). It is hard to interpret what it means to be "similar" because of their multi-dimensionality and complexity. Moreover, they might depend on the interaction type: different pressure applied when touching, velocity, slippage, or temperature. All these factors significantly complicate the clustering process and exceed the abilities of the classical methods.

Latent representations are commonly used in robotics to represent high-dimensional signals in a lower-dimensional space, making it easier for algorithms to process and understand. Additionally, using latent representation can help models generalize their knowledge and perform tasks in new environments without needing to be explicitly trained. Experiments in the chapter 5.2 show a slight improvement in the clustering performance between DEC methods working on raw data and latent vectors. On the other hand, using a SVD to preprocess signals for the classical approaches did not let them achieve comparable results.

Additionally, silhouette tests suggest that the optimal number of clusters is three. Considering that these experiments were targeted to prove or refute the existence of five tactile dimensions, one could summarize that they denied them, as there should be five of them as well. However, under the skin of a human's fingertip, an estimated 3000 touch receptors ensure deep and precise haptic sensing. On the other hand, even such an advanced touch sensor as BioTac has only 24 electrodes that measure temperature and tactile feedback. The BiGS V2 dataset included recordings from three such sensors, which results in 72 electrodes sensing the object at each grip. Even that might be insufficient to fully recreate the sense of touch in how humans perceive it. Presumably, sparse electrodes do not allow for adequate recognition of the macro and roughness of touched objects. However, there is no evidence that the rest of the tactile dimensions, i.e., temperature, hardness, and friction, cannot be sensed using these sensors. To conclude, the silhouette test partially proved the existence of tactile dimensions.

However, there is still room for improvement of sensors and algorithms to ensure accurate and robust haptic perception for a high-level understanding of touched objects and dexterous manipulation. This topic had an exploratory nature. The obtained results are an excellent starting point for further development during post-doctoral research.

Chapter 9

Terrain recognition

9.1 State of the art Haptic Transformer

Terrain classification is one of the most prevalent features of the robotic perception system of a walking machine. The proper recognition enables a successful gait adaptation and lets the robot avoid terrains that are too hazardous to traverse. However, such a system cannot be too computationally exhaustive because a typical onboard computer has limited resources.

The HAPTR and HAPTR2 are novel methods for terrain classification with transformer neural networks. The attention was put on the real-world applicability and compared our approach with multiple data-driven methods, including adapted non-deep learning (KNN-DTW, ROCKET) and deep learning models (TCN, CNN-RNN). The presented comparison took into account the accuracy of each method, the number of learnable parameters, and the inference time. Tests revealed that the HAPTR2 provides the best trade-off between the accuracy and the number of parameters directly impacting inference time. Moreover, the inference time of state-of-the-art CNN-RNN takes too long to be applied on a real robot proving the need for broader evaluation than direct accuracy measurement.

9.2 Improved robustness of a perception system

Additionally, the following work focused on the robustness of robotic perception systems and introduced the MAL in HAPTR2. By assigning weights to entire modalities (F/T, inertial sensor readings) using the dot product attention layer, the model self-attended to relevant parts of an input data stream. It resulted in increased robustness of the perception system against payload changes and deterioration of signal quality. However, the MAL was implemented as a universal, standalone module that could assign weights to any user-defined modalities, creating new opportunities for future research. Afterward, to establish a fair comparison with the current state of the art, the HAPTR2 was examined using the QCAT dataset. The results showed that the HAPTR2 outperformed the complex RNNs+FC approach [56] considering accuracy and inference time performance while having over 30× less learnable parameters.

Eventually, the findings from the following dissertation were also reflected in the most up-to-date literature on the topic and fit in with existing trends in time series recognition. The usage of convolutions together with attention modules was justified in the theoretical work [119], while similar ideas regarding transformers for time-series present concurrent study presented in [117].

Chapter 10

Robust multi-modal fusion

10.1 Heterogeneous data fusion

Typically, the more complex assignment required from the robot, the more sophisticated its perception system. For example, a robot that must detect an object in the scene and apply a specified control strategy for a pick-and-place task might require data from multiple modalities sequentially – visual detection and haptic manipulation. However, looking at both parts of the mentioned task, each can be split into more steps. A visual approach might utilize RGB images, point clouds, multispectral images, or thermal images. Haptic manipulation would need to process F/T signals, inertial feedback, or highly sophisticated sensory feedback from modern haptic sensors such as GelSight or BioTac. One can observe that using many modalities at once might be troublesome, as the data often is not homogeneous, thus requiring different methods to process them. The experimental verification focused on three tasks: a grasp outcome prediction, texture recognition, and multi-label classification of haptic adjectives.

The following work compares four state-of-the-art fusion methods with further analysis of obtained outcomes. It analyzes the existence and influence of the performance of so-called leading modalities. In the experiments, one modality always had a superior impact on obtained results. Because of that fact, the quality of selected methods decreased when that modality was noised or zeroed, and deep neural networks were prone to overfit these modalities.

10.2 Data degradation robustness

The experiments examined fusion methods in possible scenarios of input data degradation that might occur in real life, e.g., a sensor turn-off or measurement noise. Finally, the influence of the data augmentation technique on the predictive capabilities of tested methods was tested and again evaluated their robustness on noise added to the leading modality and its zeroing. To build reliable autonomous systems, we must focus more on the robustness of our data fusion methods introducing adaptive behavior in the data deterioration scenario. A solution for such a problem would be the MAL presented in the following dissertation built on top of attention modules.

Chapter 11

Final remarks

11.1 Note from the author

Throughout my Ph.D., I have significantly contributed to robotics and artificial intelligence, aiming at haptic perception for manipulators and walking robots. My research's most important scientific achievements include the publication of datasets and the implementation of novel neural networks that enable embodied and artificial intelligence to estimate the stiffness of materials using inertial sensors. This connection was a novelty in soft robotics because the field primarily focused on machines' mechanical design, not the algorithms themselves. Additionally, I have proved tactile dimensions in haptic data using novel and open-source datasets, which constitutes a sound introduction to the further development of touch recognition systems in an unsupervised manner, which is still a heavily underestimated topic in the literature. I have also introduced a novel deep learning method for the terrain classification of a walking robot, which uses attention layers to improve performance compared to traditional methods and other deep learning models. The proposed method utilizes inter and intra-modalities relationships to push further the accuracy of the classifier and improve its robustness against the noise. Weights produced by the attention layer might directly indicate the importance of the modalities of the multi-dimensional signal, improving the method's interpretability. Finally, I have thoroughly evaluated noise types and other degradation scenarios in time series classification using deep neural networks. This contribution is a crucial step toward developing more robust and accurate models, and my research improved the existing state of the art. These achievements would be incomplete if I did not embed them among existing solutions to the problems raised. In this way, my work's scientific value can support other researchers in their work.

However, in addition to the work presented here, I have also contributed to the field through presentations of my work and that of my colleagues at the Institute of Robotics and Machine Intelligence, both locally and globally. I was a co-author of the work "What am I touching? Learning to classify terrain via haptic sensing" [75] at the prestigious International Conference on Robotics and Automation held in Montreal (ICRA, 2019) and "Robotic touch: Classification of materials for manipulation and walking" [61] at the International Conference on Soft Robotics held in Seoul (RoboSoft, 2019). Furthermore, throughout my Ph.D., I have worked as a researcher in multiple scientific projects in the field of robotics at the national and international levels, including

- subTerranean Haptic INvestiGator (THING, Horizon 2020): This project aimed to improve a perception system for robotic exploration of subterranean environments. I contributed to the design and implementation of the terrain classification methods, as well as the analysis of the data.
- Robotic tEchnologies for the Manipulation of cOmplex DeformablE Linear objects (REMODEL, Horizon 2020): This project focused on developing novel robotic technologies to manipulate deformable linear objects, such as cables and wires. I worked on algorithms utilizing prototypes of novel haptic sensors.
- Perception and control in a robotic task manipulation of flexible objects (LIDER program by The National Centre for Research and Development in Poland): This project aimed to improve the perception and control of manipulators in tasks involving flexible objects. I contributed to the development and testing of the algorithms and the analysis of the results.

I also did an internship at the École Polytechnique Fédérale de Lausanne (EPFL), where I was a part of the prominent Biorobotics Laboratory (BioRob) and worked on haptic classification using a quadrupedal robot. Collaborating with researchers from various countries and universities, including Eidgenössische Technische Hochschule Zürich (ETH), EPFL, the University of Edinburgh, and Oxford University, has provided me with unique experiences and invaluable opportunities for growth and development in my research skills.

Overall, my Ph.D. experience has provided me with diverse and challenging research opportunities and valuable collaborations with leading researchers in the field. These experiences have prepared me well for a successful robotics research and development career.

11.2 List of publications

When writing the following thesis, I authored and co-authored 15 scientific articles on robotics at international conferences and in scientific journals. My publications reached 60 citations according to the Scopus database, reaching an h-index equal to 5 and 96 citations according to Google Scholar, yielding an h-index equal to 6.

1. HAPTR2: Improved Haptic Transformer for Legged Robots' Terrain Classification – M. Bednarek, M. R. Nowicki, K. Walas, Robotics and Autonomous Systems – Selected papers from the 10th European Conference on Mobile Robots, 2022 (IF 2022: 3.7)

2. Fast Haptic Terrain Classification for Legged Robots Using Transformer – M. Bednarek, M. Łysakowski, J. Bednarek, M. R. Nowicki, and K. Walas, European Conference on Mobile Robots, 2021

3. Tell me, what do you see? - interpretable classification of wiring harness branches with deep neural networks - P. Kicki, M. Bednarek, P. Lembicz, G. Mierzwiak, A. Szymko, M. Kraft, & K. Walas, Sensors Journal, 2021 (IF 2021: 3.847)

4. Gaining a Sense of Touch Object Stiffness Estimation Using a Soft Gripper and Neural Networks – M. Bednarek, P. Kicki, J. Bednarek, K. Walas, Electronics Journal, 2021 (IF 2021: 2.690)

5. On Robustness of Multi-Modal Fusion—Robotics Perspective – M. Bednarek, P. Kicki, K. Walas, Electronics Journal, 2020 (IF 2021: 2.690)

6. Comparative Assessment of Reinforcement Learning Algorithms in the Task of Robotic Manipulation of Deformable Linear Objects – M. Bednarek, K. Walas, International Conference on Robotics and Automation Engineering (ICRAE), Singapore 2019

7. Robotic Manipulation of Elongated and Elastic Objects, P. Kicki, M. Bednarek, K. Walas, Signal Processing: Algorithms, Architectures, Arrangements, and Applications, Poznan 2019

8. What am I touching? Learning to classify terrain via haptic sensing – J. Bednarek, M. Bednarek, L. Wellhausen, M. Hutter, K. Walas, International Conference on Robotics and Automation (ICRA), Montreal 2019

9. Robotic Touch: Classification of Materials for Manipulation and Walking, J. Bednarek, M. Bednarek, P. Kicki, K. Walas, International Conference on Soft Robotics, Seoul 2019

10. Measuring Bending Angle and Hallucinating Shape of Elongated Deformable Objects – P. Kicki, M. Bednarek, K. Walas, Humanoids, Beijing 2019

11. Spatial Transformations in Deep Neural Networks – M. Bednarek, K. Walas, Signal Processing: Algorithms, Architectures, Arrangements, and Applications, Poznan 2018

12. Simulated Local Deformation & Focal Length Optimisation for Improved Template- Based Non-Rigid Object 3D Reconstruction – M. Bednarek, K. Walas, Signal Processing: Algorithms, Architectures, Arrangements, and Applications, Poznan 2018

13. Methods of Enriching the Flow of Information in The Real-Time Semantic Segmentation Using Deep Neural Networks – J. Bednarek, K. Piaskowski, M. Bednarek, Signal Processing: Algorithms, Architectures, Arrangements, and Applications, Poznan 2018

14. Local Descriptors Robust to Out-of-Plane Rotations – M. Bednarek, K. Walas, Signal Processing: Algorithms, Architectures, Arrangements, and Applications, Poznan 2017

15. Comparison of Visual Descriptors for 3D Reconstruction of Non-Rigid Planar Surfaces – M. Bednarek, International Conference on Image Processing & Communications, Bydgoszcz 2017

Bibliography

[1] B. S. Homberg, R. K. Katzschmann, M. R. Dogar, and D. Rus, "Haptic identification of objects using a modular soft robotic gripper," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1698–1705.

[2] Y. Chebotar, K. Hausman, Z. Su, A. Molchanov, O. Kroemer, G. Sukhatme, and S. Schaal, "Bigs: Biotac grasp stability dataset," in *ICRA 2016 Workshop on Grasping and Manipulation Datasets*, 05 2016.

[3] J. J. Culbertson, Heather; Lopez Delgado and K. J. Kuchenbecker, "The penn haptic texture toolkit for modeling, rendering, and evaluating haptic virtual textures," 2014.

[4] Y. Gao, L. A. Hendricks, K. J. Kuchenbecker, and T. Darrell, "Deep learning for tactile understanding from visual and haptic data," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 536–543.

[5] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ser. ICML'16. JMLR.org, 2016, p. 478–487.

[6] M. Hutter *et al.*, "Anymal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 38–44.

[7] P. Arm, R. Zenkl, P. Barton, L. Beglinger, A. Dietsche, L. Ferrazzini, E. Hampp, J. Hinder, C. Huber, D. Schaufelberger, F. Schmitt, B. Sun, B. Stolz, H. Kolvenbach, and M. Hutter, "Spacebok: A dynamic legged robot for space exploration," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6288–6294.

[8] R. Pfeifer, M. Lungarella, and F. Iida, “Self-organization, embodiment, and biologically inspired robotics,” *Science (New York, N.Y.)*, vol. 318, pp. 1088–93, 12 2007.

[9] L. U. Odhner, R. R. Ma, and A. M. Dollar, “Open-loop precision grasping with underactuated hands inspired by a human manipulation strategy,” *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 625–633, 2013.

[10] S. Li, J. J. Stampfli, H. J. Xu, E. Malkin, E. V. Diaz, D. Rus, and R. J. Wood, “A vacuum-driven origami "magic-ball" soft gripper,” in *International Conference on Robotics and Automation, ICRA 2019, Montreal, QC, Canada, May 20-24, 2019*, 2019, pp. 7401–7408.

[11] M. Manti, T. Hassan, G. Passetti, N. D’Elia, C. Laschi, and M. Cianchetti, “A bioinspired soft robotic gripper for adaptable and effective grasping,” *Soft Robotics*, vol. 2, 08 2015.

[12] A. Atalay, V. Sanchez, O. Atalay, D. Vogt, F. Haufe, R. J. Wood, and C. J. Walsh, “Batch fabrication of customizable silicone-textile composite capacitive strain sensors for human motion tracking,” *Advanced Materials Technologies*, 2017.

[13] C. Chorley, C. Melhuish, T. Pipe, and J. Rossiter, “Development of a tactile sensor based on biologically inspired edge encoding,” *Advanced Robotics, ICAR*, 2009.

[14] *A Lower Limb Prosthesis Haptic Feedback System for Stair Descent*, ser. Frontiers in Biomedical Devices, vol. 2017 Design of Medical Devices Conference, 04 2017, v001T05A004.

[15] M. Strese, J. Y. Lee, C. Schuwerk, Q. Han, H. G. Kim, and E. Steinbach, “A haptic texture database for tool-mediated texture recognition and classification,” in *2014 IEEE International Symposium on Haptic, Audio and Visual Environments and Games, HAVE 2014 - Proceedings*, 2014, pp. 118–123.

[16] A. Tulbure and B. Baeuml, “Superhuman Performance in Tactile Material Classification and Differentiation with a Flexible Pressure-Sensitive Skin,” in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids) Beijing,*, 2018, pp. 306–313.

[17] M. Kerzel, M. Ali, H. G. Ng, and S. Wermter, "Haptic material classification with a multi-channel neural network," *Proceedings of the International Joint Conference on Neural Networks*, vol. 2017-May, pp. 439–446, 2017.

[18] R. Buchanan, J. Bednarek, M. Camurri, M. R. Nowicki, K. Walas, and M. Fallon, "Navigating by touch: Haptic monte carlo localization via geometric sensing and terrain classification," *Robotics and Autonomous Systems (accepted)*, 2021.

[19] N. Roy, I. Posner, T. Barfoot, P. Beaudoin, Y. Bengio, J. Bohg, O. Brock, I. Depatie, D. Fox, D. Koditschek, *et al.*, "From machine learning to robotics: Challenges and opportunities for embodied intelligence," *arXiv preprint arXiv:2110.15245*, 2021.

[20] P. Wang, L. Wang, R. Chen, J. Xu, J. Xu, and M. Gao, "Overview and outlook on railway track stiffness measurement," *Journal of Modern Transportation*, vol. 24, no. 2, pp. 89–102, Jun 2016.

[21] A. Matsubara, T. Yamazaki, and S. Ikenaga, "Non-contact measurement of spindle stiffness by using magnetic loading device," *International Journal of Machine Tools and Manufacture*, vol. 71, pp. 20–25, 2013.

[22] X. Li and B. Bhushan, "A review of nanoindentation continuous stiffness measurement technique and its applications," *Materials Characterization*, vol. 48, no. 1, pp. 11–36, 2002.

[23] O. Sul, E. Choi, and S.-B. Lee, "A portable stiffness measurement system," *Sensors (Basel, Switzerland)*, vol. 17, no. 11, p. 2686, Nov 2017, 29160821[pmid].

[24] A. Marter, A. Dickinson, F. Pierron, and M. Browne, "A practical procedure for measuring the stiffness of foam like materials," *Experimental Techniques*, vol. 42, no. 4, pp. 439–452, Aug 2018.

[25] M. Petrů and O. Novák, "Measurement and numerical modeling of mechanical properties of polyurethane foams," in *Aspects of Polyurethanes*, F. Yilmaz, Ed. Rijeka: IntechOpen, 2017, ch. 4.

[26] G. Santaera, E. Luberto, A. Serio, M. Gabiccini, and A. Bicchi, "Low-cost, fast and accurate reconstruction of robotic and human postures via imu measurements," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2728–2735.

[27] F. Coutinho and R. Cortesão, "Online stiffness estimation for robotic tasks with force observers," *Control Engineering Practice*, vol. 24, pp. 92–105, 2014.

[28] G. Hattori and A. L. Serpa, "Contact stiffness estimation in ansys using simplified models and artificial neural networks," *Finite Elements in Analysis and Design*, vol. 97, pp. 43–53, 2015.

[29] N. Wettels, V. Santos, R. Johansson, and G. Loeb, "Biomimetic tactile sensor array," *Advanced Robotics*, vol. 22, pp. 829–849, 08 2008.

[30] K. Chin, T. Hellebrekers, and C. Majidi, "Machine learning for soft robotic sensing and control," *Advanced Intelligent Systems*, vol. 2, no. 6, p. 1900171, 2020.

[31] J. Zimmer, T. Hellebrekers, T. Asfour, C. Majidi, and O. Kroemer, "Predicting grasp success with a soft sensing skin and shape-memory actuated gripper," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 7120–7127.

[32] A. Al-Ibadi, S. Nefti-Meziani, and S. Davis, "Controlling of pneumatic muscle actuator systems by parallel structure of neural network and proportional controllers (pnnp)," *Frontiers in Robotics and AI*, vol. 7, p. 115, 2020.

[33] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 124–134, 2019.

[34] J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, "Soft robot control with a learned differentiable model," in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, 2020, pp. 417–423.

[35] N. Rotella, S. Mason, S. Schaal, and L. Righetti, "Inertial sensor-based humanoid joint state estimation," 2016.

[36] A. Ancillao, S. Tedesco, J. Barton, and B. O'Flynn, "Indirect measurement of ground reaction forces and moments by means of wearable inertial sensors: A systematic review," *Sensors (Basel, Switzerland)*, vol. 18, no. 8, p. 2564, Aug 2018, 30081607[pmid].

[37] N. Slobodenyuk, Y. Jraissati, A. Kanso, L. Ghanem, and I. Elhajj, "Cross-modal associations between color and haptics," *Attention, perception & psychophysics*, vol. 68, 03 2015.

[38] M. Ji, L. Fang, H. Zheng, M. Strese, and E. Steinbach, "Preprocessing-free surface material classification using convolutional neural networks pretrained by sparse autoencoder," in *IEEE International Workshop on Machine Learning for Signal Processing*, Sep 2015.

[39] ——, "Preprocessing-free surface material classification using convolutional neural networks pretrained by sparse Autoencoder," *IEEE International Workshop on Machine Learning for Signal Processing, MLSP*, 2015.

[40] Z. Erickson, S. Chernova, and C. C. Kemp, "Semi-supervised haptic material recognition for robots using generative adversarial networks," 2017.

[41] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 2672–2680.

[42] Z. Erickson, H. M. Clever, G. Turk, C. K. Liu, and C. C. Kemp, "Deep haptic model predictive control for robot-assisted dressing," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018.

[43] P. Lakhan, N. Banluesombatkul, V. Changniam, R. Dhithijaiyratn, P. Leelaarporn, E. Boonchieng, S. Hompoonsup, and T. Wilaiprasitporn, "Consumer grade brain sensing for emotion recognition," *IEEE Sensors Journal*, vol. 19, no. 21, pp. 9896–9907, 2019.

[44] Y. Song, Y. Wang, and J. Viventi, "Unsupervised learning of spike patterns for seizure detection and wavefront estimation of high resolution micro electrocorticographic (μ ecog) data," *IEEE Transactions on NanoBioscience*, vol. 16, no. 6, pp. 418–427, 2017.

[45] D. Meli and P. Fiorini, "Unsupervised identification of surgical robotic actions from small non-homogeneous datasets," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, p. 8205–8212, Oct 2021.

[46] F. Despinoy, D. Bouget, G. Forestier, C. Penet, N. Zemiti, P. Poignet, and P. Jannin, "Unsupervised trajectory segmentation for surgical ges-

ture recognition in robotic training," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 6, pp. 1280–1291, 2016.

[47] D. Zhao, B. Si, and F. Tang, "Unsupervised feature learning for visual place recognition in changing environments," in *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, pp. 1–8.

[48] P. Yin, L. Xu, Z. Liu, L. Li, H. Salman, Y. He, W. Xu, H. Wang, and H. Choset, "Stabilize an unsupervised feature learning for lidar-based place recognition," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1162–1167.

[49] S. Lowry and M. J. Milford, "Supervised and unsupervised linear learning techniques for visual place recognition in changing environments," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 600–613, 2016.

[50] S. Schubert, P. Neubert, and P. Protzel, "Unsupervised learning methods for visual place recognition in discretely and continuously changing environments," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4372–4378.

[51] M. R. Loghmani, L. Robbiano, M. Planamente, K. Park, B. Caputo, and M. Vincze, "Unsupervised domain adaptation through inter-modal rotation for rgb-d object recognition," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6631–6638, 2020.

[52] S. Okamoto, H. Nagano, and Y. Yamada, "Psychophysical dimensions of tactile perception of textures," *IEEE Transactions on Haptics*, vol. 6, no. 1, pp. 81–93, 2013.

[53] B. A. Richardson and K. J. Kuchenbecker, "Improving haptic adjective recognition with unsupervised feature learning," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3804–3810.

[54] ——, "Learning to predict perceptual distributions of haptic adjectives," *Frontiers in Neurorobotics*, vol. 13, 2019.

[55] R. Corcodel, S. Jain, and J. van Baar, "Interactive tactile perception for classification of novel object instances," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9861–9868.

[56] A. Ahmadi, T. Nygaard, N. Kottege, D. Howard, and N. Hudson, “Semi-Supervised Gated Recurrent Neural Networks for Robotic Terrain Classification,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1848–1855, 2021.

[57] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for games, robotics and machine learning,” http://pybullet.org, 2016–2021.

[58] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, Sep 2004, pp. 2149–2154.

[59] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[60] M. Schuster and K. Paliwal, “Bidirectional recurrent neural networks,” *Trans. Sig. Proc.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[61] J. Bednarek, M. Bednarek, P. Kicki, and K. Walas, “Robotic Touch: Classification of Materials for Manipulation and Walking,” in *IEEE International Conference on Soft Robotics (RoboSoft)*, 2019, pp. 527–533.

[62] A. Garcia-Garcia, B. S. Zapata-Impata, S. Orts-Escolano, P. Gil, and J. Garcia-Rodriguez, “Tactilegcn: A graph convolutional network for predicting grasp stability with tactile sensors,” in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2019.

[63] P. Dallaire, P. Giguère, D. Émond, and B. Chaib-draa, “Autonomous tactile perception: A combined improved sensing and bayesian nonparametric approach,” *Robotics and Autonomous Systems*, vol. 62, no. 4, pp. 422–435, 2014.

[64] O. Kroemer, C. Lampert, and J. Peters, “Learning Dynamic Tactile Sensing With Robust Vision-Based Training,” *Robotics, IEEE Trans. on*, vol. 27, no. 3, pp. 545–557, june 2011.

[65] B. R. Corporation, “The algorithm workshop,” accessed on 15.03.2022. [Online]. Available: https://brc2.com/the-algorithm-workshop/

[66] N. X. Vinh, J. Epps, and J. Bailey, “Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance,” *Journal of Machine Learning Research*, vol. 11, no. 95, pp. 2837–2854, 2010.

[67] S. X. Yu and J. Shi, "Multiclass spectral clustering." in *ICCV*. IEEE Computer Society, 2003, pp. 313–319.

[68] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Pearson, 2018.

[69] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: An efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 103–114.

[70] R. R. Sokal and C. D. Michener, "A statistical method for evaluating systematic relationships," *University of Kansas Science Bulletin*, vol. 38, pp. 1409–1438, 1958.

[71] A. Bouman *et al.*, "Autonomous spot: Long-range autonomous exploration of extreme environments with legged locomotion," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2518–2525.

[72] A. Roennau, G. Heppner, M. Nowicki, J. Zoellner, and R. Dillmann, "Reactive posture behaviors for stable legged locomotion over steep inclines and large obstacles," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 4888–4894.

[73] D. Bellicoso *et al.*, "Perception-less terrain adaptation through whole body control and hierarchical optimization," *IEEE-RAS International Conference on Humanoid Robots*, pp. 558–564, 2016.

[74] H. Kolvenbach, C. Bärtschi, L. Wellhausen, R. Grandia, and M. Hutter, "Haptic Inspection of Planetary Soils With Legged Robots," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1626–1632, 2019.

[75] J. Bednarek, M. Bednarek, L. Wellhausen, M. Hutter, and K. Walas, "What am I touching? Learning to classify terrain via haptic sensing," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 7187–7193.

[76] M. A. Hoepflinger *et al.*, "Haptic terrain classification on natural terrains for legged robots," *Proc. of the 13th International Conference on Climbing and Walking Robots, CLAWAR 2010*, pp. 785–792, 2010.

[77] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in *Proc. of the 13th International Conference on Machine Learning*, ser. ICML’96. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996, p. 148–156.

[78] X. A. Wu, T. M. Huh, R. Mukherjee, and M. Cutkosky, “Integrated Ground Reaction Force Sensing and Terrain Classification for Small Legged Robots,” *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1125–1132, 2016.

[79] X. Li, W. Wang, and J. Yi, “Ground substrate classification for adaptive quadruped locomotion,” in *Proceedings - IEEE International Conference on Robotics and Automation*, 2017, pp. 3237–3243.

[80] X. A. Wu, T. M. Huh, A. Sabin, S. A. Suresh, and M. R. Cutkosky, “Tactile Sensing and Terrain-Based Gait Control for Small Legged Robots,” *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 15–27, feb 2020.

[81] E. Tennakoon, T. Peynot, J. Roberts, and N. Kottege, “Probe-before-step walking strategy for multi-legged robots on terrain with risk of collapse,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 5530–5536, 2020.

[82] W. Bosworth, J. Whitney, S. Kim, and N. Hogan, “Robot locomotion on hard and soft ground: Measuring stability and ground properties in-situ,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 3582–3589, 2016.

[83] S. Fahmi, G. Fink, and C. Semini, “On State Estimation for Legged Locomotion over Soft Terrain,” *IEEE Sensors Letters*, vol. 5, no. 1, pp. 1–4, 2021.

[84] S. Fahmi, M. Focchi, A. Radulescu, G. Fink, V. Barasuol, and C. Semini, “STANCE: Locomotion Adaptation over Soft Terrain,” *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 443–457, 2020.

[85] P. Filitchkin and K. Byl, “Feature-based terrain classification for LittleDog,” *IEEE International Conference on Intelligent Robots and Systems (IROS)*, no. 2, pp. 1387–1392, 2012.

[86] J. Christie and N. Kottege, “Acoustics based terrain classification for legged robots,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 3596–3603, 2016.

[87] K. Walas, "Terrain Classification and Negotiation with a Walking Robot," *Journal of Intelligent & Robotic Systems*, vol. 78, no. 3-4, pp. 401–423, 2015.

[88] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk(Predicting terrain properties from images via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, 2019.

[89] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[90] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Min. Knowl. Discov.*, vol. 29, no. 3, p. 565–592, May 2015.

[91] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[92] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with cote: The collective of transformation-based ensembles," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.

[93] J. Lines, S. Taylor, and A. Bagnall, "Hive-cote: The hierarchical vote collective of transformation-based ensembles for time series classification," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 2016, pp. 1041–1046.

[94] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, "Classification of time series by shapelet transformation," *Data Min. Knowl. Discov.*, vol. 28, no. 4, p. 851–881, July 2014.

[95] P. Schäfer, "The boss is concerned with time series classification in the presence of noise," *Data Min. Knowl. Discov.*, vol. 29, no. 6, p. 1505–1530, Nov. 2015.

[96] B. Lucas *et al.*, "Proximity forest: an effective and scalable distance-based classifier for time series," *Data Mining and Knowledge Discovery*, vol. 33, no. 3, pp. 607–635, May 2019.

[97] H. Ismail Fawaz *et al.*, “InceptionTime: Finding AlexNet for time series classification,” *Data Mining and Knowledge Discovery*, vol. 34, no. 6, pp. 1936–1962, 2020.

[98] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, inception-resnet and the impact of residual connections on learning,” in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, ser. AAAI'17. AAAI Press, 2017, p. 4278–4284.

[99] A. Dempster, F. Petitjean, and G. I. Webb, “ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels,” *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, 2020.

[100] S. Bai, J. Z. Kolter, and V. Koltun, “An empirical evaluation of generic convolutional and recurrent networks for sequence modeling,” 2018.

[101] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017.

[102] T. B. Brown *et al.*, “Language models are few-shot learners,” *CoRR*, vol. abs/2005.14165, 2020.

[103] A. Dosovitskiy *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” 2020.

[104] N. Carion *et al.*, “End-to-end object detection with transformers,” in *Computer Vision – ECCV 2020*. Cham: Springer International Publishing, 2020, pp. 213–229.

[105] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, “Learning robust perceptive locomotion for quadrupedal robots in the wild,” *Science Robotics*, vol. 7, no. 62, p. eabk2822, 2022.

[106] “Boston dynamic's website,” accessed on 03.02.2022. [Online]. Available: https://www.bostondynamics.com/

[107] “Agility robotics' website,” accessed on 03.02.2022. [Online]. Available: https://www.agilityrobotics.com/

[108] “Gost robotics' website,” accessed on 03.02.2022. [Online]. Available: https://www.ghostrobotics.io/

[109] C. Gehring, P. Fankhauser, L. Isler, R. Diethelm, S. Bachmann, M. Potz, L. Gerstenberg, and M. Hutter, "Anymal in the field: Solving industrial inspection of an offshore hvdc platform with a quadrupedal robot," in *Field and Service Robotics*, G. Ishigami and K. Yoshida, Eds. Singapore: Springer Singapore, 2021, pp. 247–260.

[110] K. Walas, D. Kanoulas, and P. Kryczka, "Terrain classification and locomotion parameters adaptation for humanoid robots using force/torque sensing," in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*, 2016, pp. 133–140.

[111] A. H. Chang, C. Hubicki, A. Ames, and P. A. Vela, "Every hop is an opportunity: Quickly classifying and adapting to terrain during targeted hopping," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3188–3194.

[112] R. Ahmadi, T. Nygaard, N. Kottege, D. Howard, and N. Hudson, "Qcat legged robot terrain classification dataset," 2020.

[113] G. Valsecchi, R. Grandia, and M. Hutter, "Quadrupedal locomotion on uneven terrain with sensorized feet," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1548–1555, 2020.

[114] T. F. Nygaard, C. P. Martin, J. Torresen, and K. Glette, "Self-modifying morphology experiments with dyret: Dynamic robot for embodied testing," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9446–9452.

[115] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[116] M. Löning *et al.*, "Sktime: A unified interface for machine learning with time series," *arXiv*, 2019.

[117] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, "A transformer-based framework for multivariate time series representation learning," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 2114–2124.

[118] M. Bednarek, M. Łysakowski, J. Bednarek, M. R. Nowicki, and K. Walas, "Fast haptic terrain classification for legged robots using transformer," in *2021 European Conference on Mobile Robots (ECMR)*, 2021, pp. 1–7.

[119] N. Park and S. Kim, "How do vision transformers work?" in *International Conference on Learning Representations*, 2022.

[120] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," 2017.

[121] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, pp. 746–748, 12 1976.

[122] T. Baltrusaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, p. 423–443, Feb. 2019.

[123] N. Fazeli, M. Oller, J. Wu, Z. Wu, J. Tenenbaum, and A. Rodriguez, "See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion," *Science Robotics*, vol. 4, p. eaav3123, 01 2019.

[124] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8943–8950.

[125] W. Yuan, S. Dong, and E. Adelson, "Gelsight: High-resolution robot tactile sensors for estimating geometry and force," *Sensors*, vol. 17, p. 2762, 11 2017.

[126] G. Izatt, G. Mirano, E. Adelson, and R. Tedrake, "Tracking objects with point clouds from vision and touch," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4000–4007.

[127] O. Mees, A. Eitel, and W. Burgard, "Choosing smartly: Adaptive multimodal fusion for object detection in changing environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Deajeon, South Korea, Oct. 2016.

[128] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4644–4651.

[129] C. Hung, J. Nieto, Z. Taylor, J. Underwood, and S. Sukkarieh, "Orchard fruit segmentation using multi-spectral feature learning," in *2013 IEEE/RSJ International Conf. on Intelligent Robots and Systems*. IEEE, 2013, pp. 5314–5320.

[130] J. Ilonen, J. Bohg, and V. Kyrki, "Fusing visual and tactile sensing for 3-d object reconstruction while grasping," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 3547–3554.

[131] D. Ramachandram and G. W. Taylor, "Deep multimodal learning: A survey on recent advances and trends," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 96–108, 2017.

[132] F. Castanedo, "A review of data fusion techniques," *The Scientific World Journal*, vol. 2013, 2013.

[133] H. Liu, F. Li, X. Xu, and F. Sun, "Multi-modal local receptive field extreme learning machine for object recognition," *Neurocomputing*, vol. 277, pp. 4–11, 2018, hierarchical Extreme Learning Machines.

[134] Z. Xiong, Y. Yuan, and Q. Wang, "Rgb-d scene recognition via spatial-related multi-modal feature learning," *IEEE Access*, vol. 7, pp. 106 739–106 747, 2019.

[135] B. Sebastian, H. Ren, and P. Ben-Tzvi, "Neural network based heterogeneous sensor fusion for robot motion planning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2899–2904.

[136] J. Choi and J.-S. Lee, "Embracenet: A robust deep learning architecture for multimodal classification," *Inf. Fusion*, vol. 51, pp. 259–270, 2019.

[137] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency, "Tensor fusion network for multimodal sentiment analysis," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sept. 2017, pp. 1103–1114.

[138] M. Hou, J. Tang, J. Zhang, W. Kong, and Q. Zhao, "Deep multimodal multilinear fusion with high-order polynomial pooling," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 12 136–12 145.

[139] Z. Liu, Y. Shen, V. B. Lakshminarasimhan, P. P. Liang, A. Zadeh, and L.-P. Morency, "Efficient low-rank multimodal fusion with modality-specific factors," in *ACL*, 2018.

[140] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, “On Combining Classifiers.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, 1998.

[141] R. Zhang, S. A. Candra, K. Vetter, and A. Zakhor, “Sensor fusion for semantic segmentation of urban scenes,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1850–1857.

[142] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, “Multimodal deep learning for robust rgb-d object recognition,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 681–687.

[143] G. M. García, E. Potapova, T. Werner, M. Zillich, M. Vincze, and S. Frintrop, “Saliency-based object discovery on rgb-d data with a late-fusion approach,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1866–1873.

[144] J. Tian, W. Cheung, N. Glaser, Y.-C. Liu, and Z. Kira, “Uno: Uncertainty-aware noisy-or multimodal fusion for unanticipated input degradation,” 2019.

[145] J. Kim, J. Koh, Y. Kim, J. Choi, Y. Hwang, and J. W. Choi, “Robust deep multi-modal learning based on gated information fusion network,” in *Computer Vision – ACCV 2018*, C. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham: Springer International Publishing, 2019, pp. 90–106.

[146] T. Kim and J. Ghosh, “On single source robustness in deep fusion models,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’ Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 4814–4825.

[147] M. Bijelic, C. Muench, W. Ritter, Y. Kalnishkan, and K. Dietmayer, “Robustness against unknown noise for raw data fusing neural networks,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2177–2184.

[148] N. Patel, A. Choromanska, P. Krishnamurthy, and F. Khorrami, “A deep learning gated architecture for ugv navigation robust to sensor failures,” *Robotics and Autonomous Systems*, vol. 116, pp. 80–97, 2019.

[149] N. Landin, J. M. Romano, W. McMahan, and K. J. Kuchenbecker, "Dimensional reduction of high-frequency accelerations for haptic rendering," in *Haptics: Generating and Perceiving Tangible Sensations*, A. M. L. Kappers, J. B. F. van Erp, W. M. Bergmann Tiest, and F. C. T. van der Helm, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 79–86.

[150] J. Bell, S. Bolanowski, and M. H. Holmes, "The structure and function of pacinian corpuscles: A review," *Progress in Neurobiology*, vol. 42, no. 1, pp. 79–128, 1994.

[151] V. Chu, I. McMahon, L. Riano, C. G. McDonald, Q. He, J. M. Perez-Tejada, M. Arrigo, T. Darrell, and K. J. Kuchenbecker, "Robotic learning of haptic adjectives through physical interaction," *Robotics and Autonomous Systems*, vol. 63, pp. 279–292, 2015, advances in Tactile Sensing and Touch-based Human Robot Interaction.

[152] K. Sechidis, G. Tsoumakas, and I. Vlahavas, "On the stratification of multi-label data," in *Machine Learning and Knowledge Discovery in Databases*, D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 145–158.

[153] B. Dasarathy, "Sensor fusion potential exploitation-innovative architectures and illustrative applications," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 24–38, 1997.

[154] M. Tan and Q. V. Le, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 2021, pp. 10 096–10 106.

[155] N. Saito, "Local feature extraction and its applications using a library of bases," Ph.D. dissertation, Yale University, 1994.

[156] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima." *CoRR*, vol. abs/1609.04836, 2016.