Bachelor Thesis

Czech
Technical
University
in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Deformable object classification through robot grasping

Michal Pliska

Supervisor: Mgr. Matěj Hoffmann, Ph.D.
Supervisor–specialist: Ing. Zdeněk Straka
Field of study: Cybernetics and Robotics
May 2021

Acknowledgements

I would like to thank Matěj Hoffmann and Zdeněk Straka for their advice and guidance. I would especially like to thank Bedřich Himmel for his help in repairing the robots. Special thanks also belong to Michal Mareš and Pavel Stoudek, as they gave me valuable advice and I built on their work.

Finally, I would like to thank my friends and family for their support.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, May 22, 2021

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 22. května 2021

Abstract

Learning about objects through manipulation importantly complements visual perception mainly to identify physical properties of objects such as stiffness, mass, or surface roughness. This holds both for human and robot perception. In this work, I study the classification of deformable objects by grasping them using four different robotic hands / grippers: Barrett Hand (3 fingers with adjustable configuration), qb SoftHand (5 fingers, 1 motor), and two industrial parallel jaw grippers (Robotiq 2F-85 and OnRobot RG6). The time series collected during object compression (and sometimes decompression) are fed into four different classifiers: k Nearest Neighbors (kNN) and LSTM applied on raw data, and kNN and SVM on features. I systematically compare the grippers' performance, together with the effects of: (i) action parameters (grasping configuration and speed of squeezing), (ii) knowledge transfer ability, and (iii) individual sensory modalities. The Robotiq 2F-85 and the Barrett Hand perform best. The OnRobot RG6 is closely in line, and qb SoftHand performs significantly worse. The 2-finger grippers thus provide a more parsimonious solution to deformable object classification relying only on the stress/strain characteristics in only 2 sensory channels (position and effort), compared to the Barrett hand with 96 tactile sensors, 3 fingertip torque sensors, and 8 joint encoders. The supervised learning problem is complemented by principal component analysis to uncover the sources of variability in the data. This work provides a unique contribution in that it deploys four different robot hands/grippers on the same datasets and systematically studies their performance. Transfer learning between different robot hands remains a future challenge.

Keywords: robotic grippers, tactile sensors, model-free object classification, object parameter extraction, LSTM network

Supervisor: Mgr. Matěj Hoffmann, Ph.D.

Abstrakt

Poznávání objektů prostřednictvím manipulace významně doplňuje vizuální vnímání a to především při určování fyzikálních vlastností objektů jako je tuhost, hmotnost nebo drsnost povrchu. Toto platí jak pro lidské, tak pro robotické vnímání. V této práci studuji klasifikaci deformovatelných objektů pomocí jejich mačkání čtyřmi různými robotickými rukama-/uchopovači: Barrett Hand (3 prsty s nastavitelnou konfigurací), qb SoftHand (5 prstů, 1 motor) a dva průmyslové grippery (Robotiq 2F-85 a OnRobot RG6). Časové řady shromážděné během stlačování (a někdy i dekomprese) objektů jsou přiváděny do čtyř různých klasifikátorů: k Nearest Neighbors (kNN) a LSTM jsou aplikované na surová data a kNN a SVM na extrahované features. Systematicky porovnávám výkonnost gripperů spolu s vlivem: (i) akčních parametrů (konfigurace uchopení a rychlost stisku), (ii) schopnost přenášet znalosti a (iii) modality jednotlivých senzorů. Nejlépe si vedou grippery Robotiq 2F-85 a Barrett Hand. V těsném závěsu za nimi je OnRobot RG6 a qb SoftHand si vede výrazně hůře. Dvouprsté grippery tak poskytují úspornější řešení klasifikace deformovatelných objektů, které se spoléhá pouze na charakteristiky napětí/deformace pouhých 2 senzorický kanálů (poloha a síla). Ve srovnání s Barrett Hand s 96 hmatovými senzory, 3 senzory točivého momentu na špičkách prstů a 8 senzory úhlu prstů. Problém učení s učitelem je doplněn analýzou hlavních komponent, která odhaluje zdroje variability v datech. Tato práce představuje jedinečný přínos v tom, že na stejných souborech dat nasazuje čtyři různé robotické ruce/chapadla a systematicky studuje jejich výkonnost. Přenos učení mezi různými robotickými rukama zůstává výzvou do budoucna.

Klíčová slova: robotická chapadla, taktilní senzory, klasifikace objektů bez modelu, zjišťování vlastností předmětu, LSTM síť

Překlad názvu: Klasifikace měkkých předmětů skrze mačkání robotickými uchopovači

Contents

Chapter 1

Introduction

1.1 Motivation

In the last few years, computer vision achieved significant progress, mainly thanks to the invention of convolution networks and new architectures like ResNet [1]. Nowadays, there are off-the-shelf methods for image classification, bounding box regression, or segmentation, such as Facebook's Detectron 2, based on Mask R-CNN architecture. However, this network primarily detects objects. For all sorts of tasks, such as waste sorting, visual information alone is not enough and one needs to gain the physical properties of an object from haptic feedback.

Object recognition using haptic exploration can also be more robust as it is insensitive to lighting conditions and attributes like color. Nevertheless, exploring objects by manipulation also brings the danger of damage. However, in some areas like the previously mentioned waste sorting, this is not a problem. Recycling is still a process relying on human labor. Plastic, paper and metal differ significantly in their material properties, and sorting them through manipulation would mean a big step forward.

So, what are the limits of modern robots' haptic feedback? Professional laboratory equipment with a high number of sensors can probably achieve good results, but what about more industrial-like grippers? Moreover, even when objects are correctly classified, we may be interested in the structure of the haptic data. Is there much hidden information, or is it a more straightforward task? I am going to find at least partial answers to these questions.

1.2 Goals

Computational costs when simulating solid deformable objects' interactions are enormous [2]. Therefore, model-free classification provides an alternative and real measurements are needed. Using four robotic grippers and two sets of objects, I will collect s sufficient number of measurements according to the standards of the machine learning community. Some of the measurements have already been collected on two robots. This is described in detail in Section 3.1.

I will design my own pipeline for data preprocessing since the samples come in series. This will consist of sample synchronisations and discarding outliers. I will also remake some gripper controlling programs to achieve higher automation in data collection.

Grippers' capabilities will be tested with the use of four classifiers, each one with a different degree of sophistication. Not only will classifiers be compared, but I will also try knowledge transfer by testing already trained models on different datasets. This may reveal the primary sources of variability in time series. Search for the main source of information in the data will also be done by ablation study. To gain additional insight into the structure of the data, principal component analysis will be used for visualization.

1.3 Related work

Sanchez et al. [2] provide a survey of robotic manipulation and sensing of deformable objects. Objects are considered deformable if they have (1) no compression strength (ropes and clothes), or (2) have a large strain[1], or present a large displacement. Additionally, a classification based on geometry is presented. In this work, I am not interested in objects of linear, planar, and cloth-like type, and will focus on triparametric objects—solid objects such as sponges or plush toys, which are also the least researched object type [2].

1.3.1 Objects classification

Specifically related to this work, Spiers et al. [3] used a single force closure grasp with an underactuated two-finger compliant gripper equipped with force sensors to classify objects of various shapes, sizes, and stiffness. The feature space used for classification via Random Forests consists only of the actuator positions and the force sensor measurements at two specific time instances (first contact and getting stuck). A feature variable's importance was calculated, and the most crucial features were determined. The gripper used and our qb SoftHand can be considered similar.

A series of neural models (MLP, CNN, LSTM) was developed for the classification of 16 objects in Bednarek et al. [4]. Not using a hand or gripper, but a spherical tip with OptoForce 3-axis optical force sensor, they processed the time series from this sensor using an LSTM employed for material classification. A similar approach is used for terrain classification on a legged robot in the same work.

[1]For linear elasticity, this implies small Young's modulus, e.g. less than 10 MPa.

1.3.2 Terrain classification

In 2014, Hoffmann et al. studied the effect of motor action, and different sensory modalities on terrain classification in a quadruped robot running with multiple gaits [5]. One approach leading to terrain classification with high accuracy was computing hand-designed features on all sensor time series and then using SVM. Because both problems (terrain and object classification) can be viewed as haptic sensing problems, I decided to try extracting the same features.

1.3.3 Prior work at FEE, CTU

In his master's thesis [6], Pavel Stoudek used three of the four robotic grippers used in this work. He also wrote control programs and collected pilot series of data. His work was an important source of information for me. However, his work focused on elasticity estimation rather than object classification.

Finally, Michal Mareš in his bachelor's thesis [7] designed sets of soft objects, wrote programs for controlling Barrett Hand, recording and processing data, used LSTM for classification, and attempted to extract properties like stiffness and density. Unfortunately, it was later found out that the ground truth values of stiffness and density are not valid, and therefore I did not try to repeat this task. This work is a natural continuation of his, which was also my primary source of information.

1.3.4 Thesis contribution

In the work of Michal Mareš [7], only one robotic hand was used for object classification. Here, I will compare more robotic hands and grippers to assess their individual performance on the same dataset. In addition, compared to Mareš [7] who used only the LSTM classifier, I will use four different classifiers to have more robust results.

Michal Mareš already tried knowledge transfer and ablation experiments, but I will deliver a more systematic overview over all hands/grippers. Finally, I will use principal component analysis to gain insight into the structure of the data. Feature-based analysis is based on results from feature-based classification.

1.4 Outline

First, I will present all objects sets, robotic hands and grippers, classifiers, method of analysis, and data preporcessing in Chapter 2.

Next, the configurations of all hands/grippers used during the measurements will be presented in Chapter 3. All created datatasets are going to be described.

In Chapter 4, all obtained results will be presented and analyzed. Summary and observations will be noted.

Finally, in Chapter 5, I am going to discuss the results, identify the limitations, and suggest possible improvements that could be made in the future.

Chapter 2

Materials and methods

In this chapter, I am going to present the basic materials and methods used in this work. I will firstly introduce soft objects and foams, which we are exploring. Then I show the grippers used for collecting data. In the last part, I will explain the methods used for classification and unsupervised analysis. Some information about object sets, Barrett Hand, and LSTM neural network are paraphrased from Michal Mareš's work [7] (it will be pointed at the appropriate places). Some information about rest of the grippers is taken from Pavel Stoudek's work [6].

2.1 Objects and foams

I used two sets of deformable objects. The first is the ordinary object set consisting of 9 mostly cuboid objects with different sizes and degrees of deformability. The second set is the polyurethane foams set consisted of 20 polyurethane foam blocks of similar size, along with reference values for elasticity and density provided by the manufacturer.

Ordinary objects set

The first set consists of mainly toy-like soft objects bought in stores; see Fig. 2.1. Cuboids are preferred over spheres, as the contact surface area does not change during deformation. In addition, all objects are highly homogeneous, as we are more interested in material properties like elasticity/stiffness (elasticity is closely related to the stiffness, but stiffness takes into account both the material's elasticity and geometry) and not the shape or mass distribution. Deformation of all objects can be studied as elastic (objects do not permanently deform during squeezing), although some objects have memory foam-like behavior (e.g., *blue die* and *blue cube*). The *yellow cube* is composed of the same material as the *yellow sponge*–it has been cut out from another exemplar of the same sponge, aiming at the dimension of the *Kinova cube*. The same is true for *blue die* and *blue cube*. Conversely, *white*

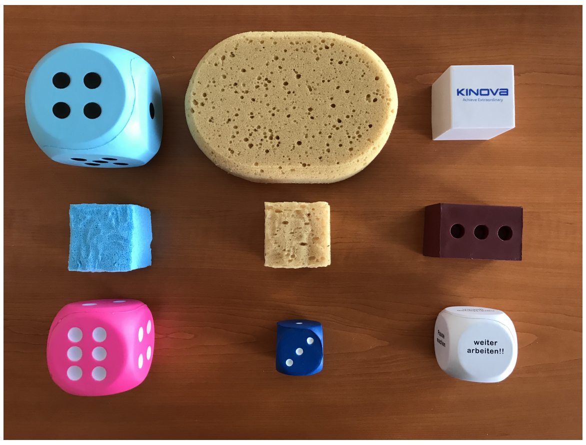


Figure 2.1: Objects. Picture from [7].

die, *kinova cube*, *yellow cube* and *blue cube* have roughly the same dimensions but different material composition and hence stiffness. The dataset has been deliberately designed to test which of the object properties are critical for model-free haptic object recognition. Source of this information is Mareš's work [7].

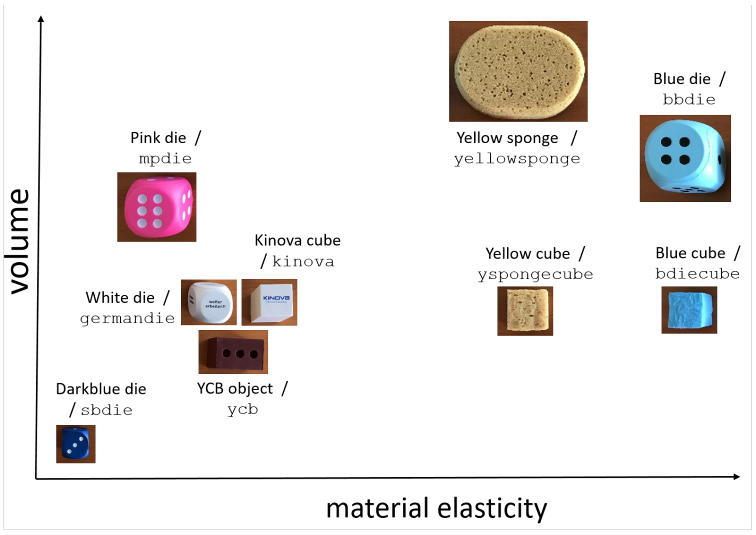


Figure 2.2: Ordinary objects set approximately spread out on the elasticity and volume axes (reference values for this object set are not available). Object names are displayed.

This set is also visualized in Fig. 2.2, approximately spread out by the stiffness/elasticity of the objects and their volume. We will later see that the distances in this graph, at least partially, correlate with misclassification rate. An overview of the object names, labels, and dimensions is in Table 2.1.

Description	Label	Dimensions [mm]
Kinova cube	kinovacube	56x56x56
Blue cube	bluecube	56x56x56
Yellow cube	ycbcube	56x56x56
Blue die	bluedie	90x90x90
White die	whitedie	59x59x59
Pink die	pinkdie	75x75x75
Darkblue die	darkbluedie	43x43x43
YCB object	ycbcube	75x50x50
Yellow sponge	yellowsponge	195x135x65

Table 2.1: Ordinary objects set – dimensions and labels. Table from [7].

2.1.1 Polyurethane foams set

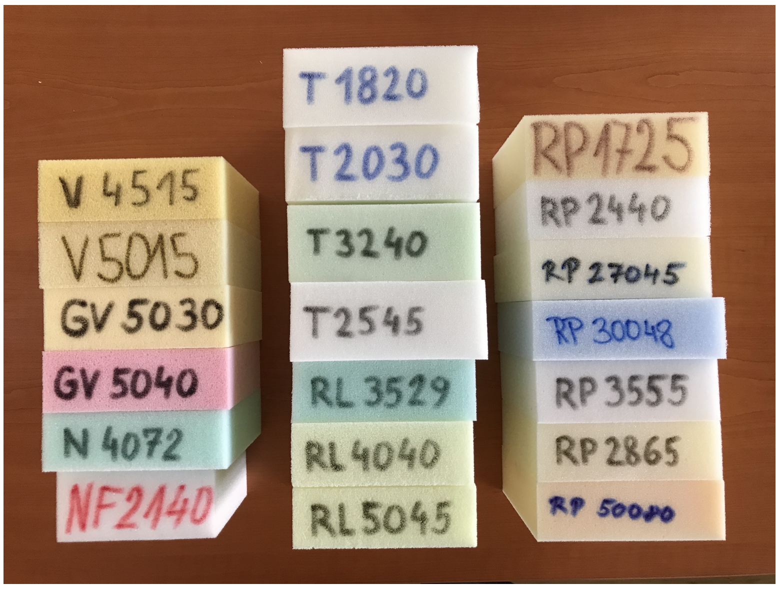


Figure 2.3: Polyurethane Foams. Picture from [7].

To complement the ordinary objects set, I tested another set of objects consisting of 20 polyurethane foams (see Fig. 2.3). These were samples provided by a manufacturer of mattresses. Their labels encode reference values for key physical properties: elasticity and density – see Table 2.2. Some of them (GV and V types) also have memory-foam-like behavior. I am going to call Compression stress value at 40% (CV_{40}) (defined in ISO standard [8]) elasticity. Source of this informations is Mareš's work [7].

2.1.2 Polyurethane foams subset

Since the polyurethane foams set is large and measuring a sufficient number of samples for each foam takes too much time, a subset of six representative

Type	Dimensions [mm]	Density [$kg \cdot m^{-3}$]	Elasticity CV_{40} [kPa]
V4515	118x120x40	45	1.5
V5015	119x120x42	50	1.5
GV5030	118x119x40	50	3.0
GV5040	118x118x39	50	4.0
N4072	118x117x37	40	7.2
NF2140	105x100x50	21	4.0
T1820	125x125x50	18	2.0
T2030	125x120x40	20	3.0
T3240	123x123x50	32	4.0
T2545	125x125x50	25	4.5
RL3529	119x118x40	35	2.9
RL4040	117x120x40	40	4.0
RL5045	118x118x39	50	4.5
RP1725	118x120x41	17	2.5
RP2440	118x120x38	24	4.0
RP27045	117x119x39	270	4.5
RP30048	123x121x39	300	4.8
RP3555	117x119x39	35	5.5
RP2865	118x118x38	28	6.5
RP50080	121x118x39	500	8.0

Table 2.2: Properties of used polyurethane foams, CV_{40} stands for "compression stress value at 40%", [8]. Table from [7].

foams was constructed to test more action parameters in a reasonable time. They are showed in Table 2.3

Type	Dimensions [mm]	Density [$kg \cdot m^{-3}$]	Elasticity CV_{40} [kPa]
V4515	118x120x40	45	1.5
NF2140	105x100x50	21	4.0
RL5045	118x118x39	50	4.5
RP1725	118x120x41	17	2.5
RP30048	123x121x39	300	4.8
RP50080	121x118x39	500	8.0

Table 2.3: Subset of six polyurethane foams.

2.2 Robot hands and grippers

Four robotic devices with different morphologies were used: two anthropomorphic robot hands and two parallel jaw 2-finger grippers. Control and dimension of feedback also vary.

2.2.1 qb SoftHand

The first gripper used is Softhand (see Fig. 2.4) made by qbrobotics [9]. The hand was mounted on UR10e robot [10], which was used only to hold the hand in position and controlled separately via the pendant. qb SoftHand is an anthropomorphic hand with five fingers the same size as a human. There is only one actuator, electric motor, which drives all joints by pulling a single string. The hand is thus highly underactuated and feedback (position, motor current) is available only from the single motor. The hand will thus passively conform to differently shaped objects. For a better idea, we can consider the following situation: If the hand starts closing empty and one finger is blocked, in the beginning, we can observe only light pressure. However, if all other fingers reach the palm and can not move any further, the pressure on the blocked finger will dramatically increase. This behavior leads to confident grips on all sorts of objects. The maximum grasping force is 62N and the nominal payload is 1.7kg.

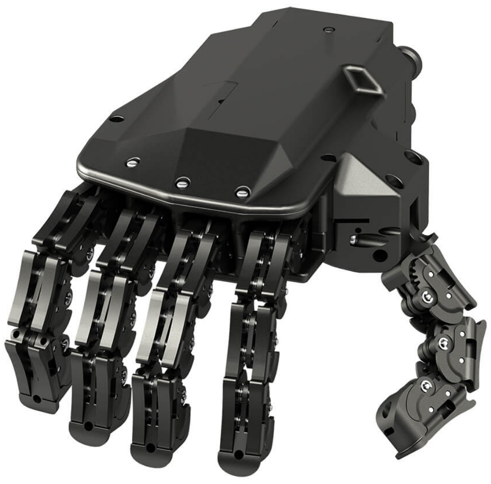


Figure 2.4: The qbrobotics SoftHand gripper. Picture from [11].

Sensors

As mentioned before, qb Softhand has only one actuator, which is the source of all signals: current and position. Since the single motor drives all coupled fingers together, these values indirectly code for the position and effort between the fingers and the object. The reading frequency is 10Hz. Position range in the interval 0-19000 without specified units and current in 0-1000 mA. Both are later in processing scaled to the <0,1> interval before any classification task or unsupervised analysis.

Control

Control via ROS was used. Official ROS package [12] allows a user to set so-called “waypoints” specifying the desired motor positions in distinct time moments. More details on the ways of control are described in the official manual [13].

```
waypoints:
-
    time: [1.0]
    joint_positions:
        qbhand1: [0.0]
-
    time: [2.25, 2.75]
    joint_positions:
        qbhand1: [0.8]
```

With this configuration, the hand will open or stay open till time 1s, then start closing and reach position 0.8 (relative motor position, the interval is <0, 1>) in time 2.25s while waiting for 0.5s. The cycle is then repeated until shutdown. To record the signals, Pavel Stoudek wrote the logging node [14] (`qb_SoftHand/qbhand_logging`) in his MSc. thesis [6].

Source of this information is Pavel Stoudek’s work [6].

2.2.2 Barrett Hand

The Barrett Hand (model BH8-282) [15] is less anthropomorphic than the qb SoftHand and has only three fingers, see Fig. 2.5. One of them is fixed, and the remaining two can rotate around the base mirroring each others angle at the base. During all measurements, the hand was mounted on the KUKA LBR iiwa arm [16], which was controlled via the pedant and served only to hold the hand in position. Each of the fingers has only one motor between the palm and the base link, which drives both joints in the finger – the mechanism is called TorqueSwitch. The behavior of Barrett Hand’s fingers is similar to the qb Softhand’s. When the base finger link can not move any further, the torque of the motor transforms to the fingertip link, enclosing the object securely in Barrett Hand. In total, there are four actuators.

Sensors

Barrett Hand has 107 sensors. Of all eight Barrett Hand’s joints, each delivers its position information (joint angle), but only four are actuated. In addition, each fingertip has a torque sensor, three in total. There is also a 6-axis torque/force sensor in the base, which will not be used in this work. What makes the Barrett Hand unique are the tactile sensors located underneath

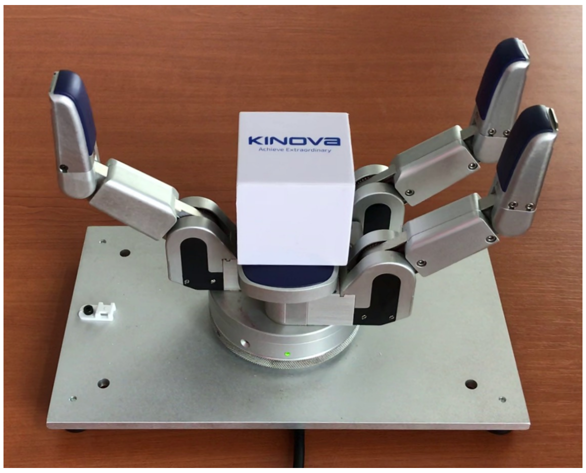
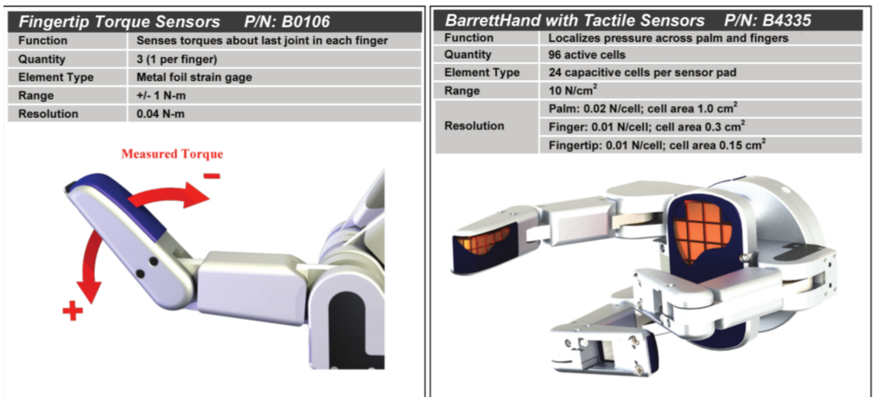


Figure 2.5: The Barrett Hand. Picture from [7].



Fingertip Torque Sensors	P/N: B0106
Function	Senses torques about last joint in each finger
Quantity	3 (1 per finger)
Element Type	Metal foil strain gage
Range	+/- 1 N-m
Resolution	0.04 N-m

Measured Torque
−
+

BarrettHand with Tactile Sensors	P/N: B4335
Function	Localizes pressure across palm and fingers
Quantity	96 active cells
Element Type	24 capacitive cells per sensor pad
Range	10 N/cm^2
Resolution	Palm: 0.02 N/cell; cell area 1.0 cm^2
	Finger: 0.01 N/cell; cell area 0.3 cm^2
	Fingertip: 0.01 N/cell; cell area 0.15 cm^2

Figure 2.6: Barrett Hand – fingertip torque and tactile sensors. Fig. from [17].

the blue plastic covers. Each of the tactile pads includes 24 capacitive cells of various surface areas, as seen in Figure Fig. 2.6.

Default recording frequency is 200 Hz, but the effective rate of tactile sensor readings was around 25 Hz. As some communication errors occurred when using the default frequency, 40 Hz was used.

Control

All hardware necessary for hand operation is enclosed in the Barrett Hand itself. The low-level messages communicate via the CAN bus protocol. Two control modes are possible: position and velocity, both controlled by PID regulators. Velocity control suits my purpose better.

I developed my own package [14] (`BarrettHand/2021-MichalPliska/Control/`) based on Michal Mareš's work. His work, explained in his bachelor thesis, stands on the following ROS packages supplied to us by Robotnik: `barrett_hand` [18], `bhand_controller` [17] and `rqt_bhand` [19]. My package builds on two main scripts:

1. Improved Mareš's script for squeezing, now also saving time of events like: start of a script, start of squeezing, hand stopping and the reason for stopping (maximal position, pressure threshold or getting stuck).

2. Script for calling previous one, which executes only one squeeze. It is possible to set the order of objects, a number of series, and a number of squeezes in one series per object. This script also saves progress, and work can be divided into more days, which is absolutely necessary since measuring a reasonable number of samples (60 squeezes per object) takes a lot of time (2 days). Discarding unsuccessful measurements after each squeeze is also implemented.

Source of this informations is Michal Mareš's work [7].

2.2.3 OnRobot RG6

The third gripper (see Fig. 2.7 and the first industrial-like is OnRobot RG6 [20]. It is a collaborative gripper with a 160mm stroke, easily changeable fingertips, adjustable force, and gripper status feedback (digital and analog). The gripper fingertips surface area is 866 mm^2. During the measuring, the gripper was mounted on a UR10e manipulator similarly to the qb Softhand.

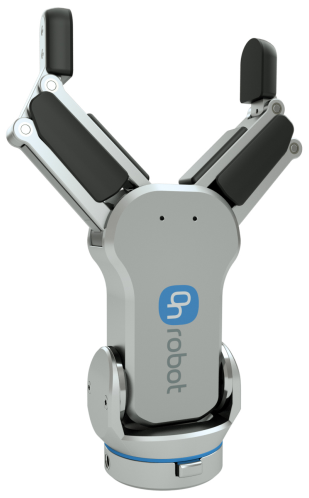


Figure 2.7: OnRobot RG6 gripper. Picture from [21].

Sensors

OnRobot RG6 has two sensor channels: gripper position (a gap between jaws) and the force between jaws in N.

Control

Continuous closing with different speeds while recording the force feedback is not possible. Instead, the gripper was commanded to close until a certain force threshold is reached. Then, the threshold was incrementally increased.

I was not doing any measurement with this robot myself and I was using data collected by Pavel Stoudek [22]. Instead, I only processed them in a different manner, which will be explained in 3.2.1.

Source of this information is Pavel Stoudek's work [6].

2.2.4 Robotiq 2F-85

The last gripper (see Fig. 2.8) used was Robotiq 2F-85 [23]. This gripper has two fingers with 85mm stroke and offers a grip force from 20N to 235N. The fingertips dimensions can be approximated with a rectangle of 37.5×22 mm. Velocity control was used for squeezing. This is quoted in % of the maximum closing speed. The gripper was mounted on a Kinova Gen3 [24] manipulator.

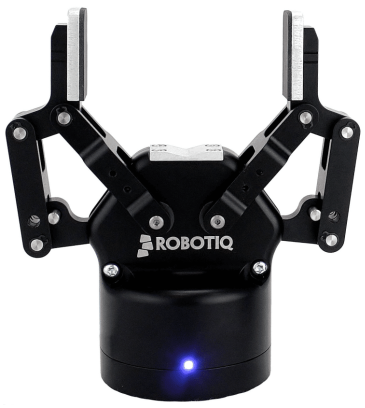


Figure 2.8: The Robotiq 2F-85 gripper. Picture from [25].

Sensors

The Robotiq 2F-85 gripper can provide various feedback options like motor temperature and voltage. However, for object classification, we used two channels: gripper position (gap between jaws 0–85 cm) and motor current (A).

Control

It is possible to control the gripper's speed. I was not doing any measurement on this robot directly since some data [26] was already measured. Instead, I only processed them in a different manner, which will be explained later on.

Source of this information is Pavel Stoudek's work [6].

2.2.5 Gripper comparison

In this section, I will compare all grippers—see Fig. 2.3 for an overview.

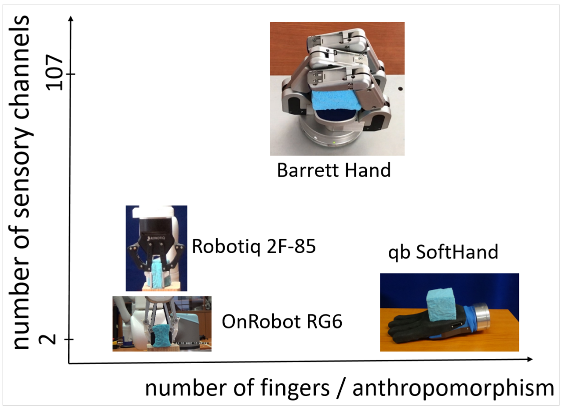


Figure 2.9: Robot hands and grippers. Two parallel jaw / two-finger grippers were employed: Robotiq 2F-85 and OnRobotRG6, with gripper position and effort (current/force) feedback. The qb Soft Hand has five fingers but only one motor and its position and current as feedback. The Barrett hand has three fingers that can be rotated around the wrist, 96 tactile sensors, 3 fingertip torque sensors, and 8 joint encoders.

Three of the four hands/grippers have only two sensory channels; the Barrett Hand has 107. All grippers have some type of position sensor. The qb SoftHand measures the position of its single motor; the Barrett Hand has angular sensors on every joint. The OnRobot RG6 and Robotiq 2F-85 measure width (gap between jaws).

The qb SoftHand and Robotiq 2F-85 measure the motor current, while the Robotiq 2F-85 measures force. The Barrett Hand has a variety of tactile sensors and also measures the torque at each finger tip.

2.3 Objects classification and unsupervised analysis of data

The first task is the classification of deformable objects using sensory time series from the grippers during compression (squeezing) and possibly decompression of the objects. I will use multiple different classifiers and compare results. The second task is the unsupervised analysis of data as an attempt to get more insight into the data's structure.

2.3.1 Objects classification

Generally, we have two options when classifying time series:

- Use raw time series. Channels then can be concatenated or used as many-dimensional arrays.
- Extract features from all channels and concatenate. Each measurement (originally the time series) is then a point in a multidimensional space. Temporal character of the data is suppressed.

I used four classifiers: K-Nearest Neighbors with time series, K-Nearest Neighbors with features, SVM with features, and LSTM with time series. The first was used as the baseline. The second uses hand-designed features to deal with the time series characters of the data and varying lengths. Its main purpose is to reveal the quality of features later used in the unsupervised analysis. As state of the art before the boom of neural networks, SVM was picked. It also uses features and is more sophisticated. This task is very interesting since fitting SVM even with the employed grid search over hyperparameters takes 10 minutes on a personal computer, unlike searching for the optimal LSTM model, which takes approximately three days on GPU grid. As nowadays state of the art, LSTM was trained to see the maximal potential of classifying soft objects with robotics grippers.

Problem formulation

Consider the input space $X = \mathbb{R}^{s \times n}$ of all possible measurements, where s is the number of sensory channels and n number of measurements (timesteps). Consider output space $Y = \{0, 1, \dots, C\}$ of labels, where $C + 1$ is the number of object categories. We are searching for a function g which will map an input vector $\mathbf{x} \in X$ to a label $y \in Y$.

$$g : X \to Y, \tag{2.1}$$

This problem formulation is the same as in Michal Mareš work [7].

As a metric for classifiers, accuracy, defined as the number of correctly classified measurements divided by the total number of measurements was used.

K-Nearest Neighbors Algorithm

K-nearest neighbors algorithm (kNN) is a simple algorithm that stores all available classified cases and classifies the new ones based on similarity measurement [27]. It is often used as a baseline classification algorithm with low generalizing capacity.

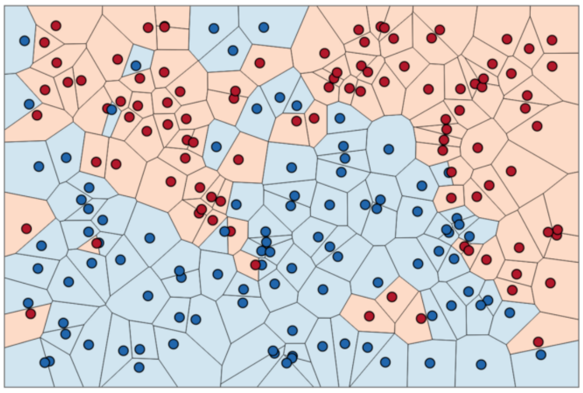


Figure 2.10: Visualisation of kNN decision process. K is here set to one. Picture from [27].

With kNN, two tuning decisions are needed. The number of nearest neighbors to form the classification rule and the distance metric. As we use kNN as the baseline classifier, I chose Euclidean distance as a metric. K is determined empirically by testing the accuracy on the validation sets of all datasets. K with the highest average accuracy on all validation sets is then chosen. This means that each gripper has its own K for all created datasets.

I used the implementation from the popular Python data science package Scikit-learn [28]. My code with dataset loading, searching for the optimal K and plotting all results can be viewed here [14] in `*robot*/*gripper*/Classification/NearestNeighbors/`.

K-Nearest Neighbors Algorithm with features

All features are computed for each sensory channel and concatenated in one vector. This vector now represents the original sample. I implemented feature extraction in pythons numpy [29]. Here is the list of them:

- Minimum value
- Mean
- Skewness
- Maximum value
- Kurtosis
- Median

- Standard deviation
- Maximum and Mean values of the first and second differences
- Approximation of the integral of values via a simple sum of values
- Amplitude of the Hilbert transform of the values

Feature	Formula
Mean value	$\overline{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$
Standard deviation	$\sigma = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(x_i - \overline{x})^2}$
Kurtosis	$\mathrm{K} = \frac{1}{N}\sum_{i=1}^{N} \frac{(x_i - \overline{x})^4}{\sigma^4}$
Skewness	$\mathrm{Sk} = \frac{1}{N}\sum_{i=1}^{N} \frac{(x_i - \overline{x})^3}{\sigma^3}$
Integral aproximation	$I = \sum_{i=1}^{N} x_i$

Table 2.4: Time domain features

Later, I found out that the derivation features (first and second differences) degrade the classification performance on at least two grippers, and I stopped using them.

Features values vary in different intervals. Therefore, scaling (or normalization) is needed. I use the scaler from Scikit-learn package [30], which sets the mean and standard deviation values across all dimensions of the feature vector to zero and one. The rest of the method was implemented the same as in the previous part 2.3.1. My code on dataset loading and features extraction, searching for optimal k, and plotting all results can be viewed here [14] in `*robot*/*gripper*/Classification/Nearest~Neighbors~with~features/`.

Support Vector Machine (SVM)

Support vector machine is a more sophisticated tool primarily designed for binary classification in N-dimensional space (N-Number of features), see [31]. The first principle is searching for the optimal separating hyperplane. This is done by maximizing the margin (margin is the minimal distance from points from both classes; the points closest to the hyperplane are called support vectors). The second principle is the use of kernel transformations. This allows separating even classes which are normally not linearly separable.

There are three main hyperparameters that can be configured:

1. Kernel, which defines the transformation of data to the space with higher dimensionality.
2. C, which controls the trade-off between smooth decision boundary and classifying training points correctly. A large value of C means getting more training points correctly.
3. γ, which defines how far the influence of a single training example reaches. If it has a low value, it means that every point has a far reach, and

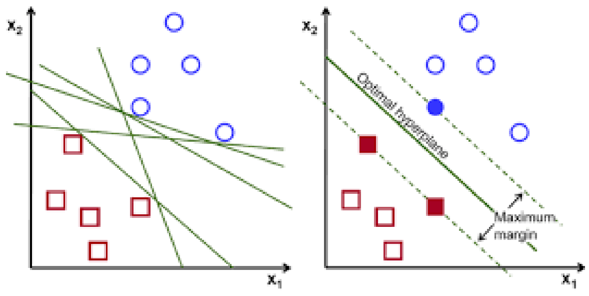
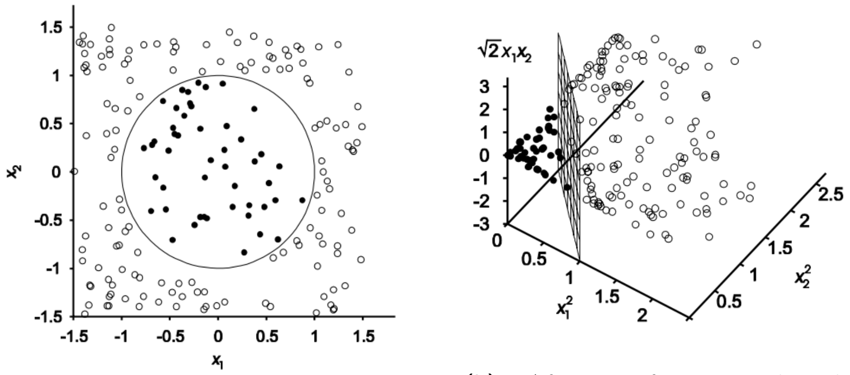


Figure 2.11: This picture visualizes the idea of maximal margin in SVM. There are infinitely many separating hyperplanes. However, optimal is only one. Fully colored markers are called support vectors. Picture from [31].



(a) : Data in the plane. We can see that classes are linearly non-separable.

(b) : After transformation, data dimensionality increased, and they can now be separated with a linear classifier.

Figure 2.12: Illustration of kernel transformation (can also be seen as increasing dimensions). Picture from [32].

conversely high value of γ means that every point has a close reach. It can be used only with rbf kernel.

Since there is no analytical rule how to choose an optimal setting, I used grid search. Features were computed the same as before. I used an implementation of SVM from the popular Python data science package Scikit-learn [33]. This package also allows defining a grid search with the use of cross-validation [34]. My code with dataset loading and features extraction, grid search and plotting all results can be viewed here[14] in `*robot*/*gripper*/Classification/SVM/`.

Long short-term memory neural network

LSTM is a type of recurrent neural network (RNN), and it is a popular option for sequence learning, such as text or speech recognition and translation [35]. The general architecture of an LSTM cell is shown in Figure 2.13. My code

[14](`*robot*/*gripper*/Classification/LSTM/`) builds on Michal Mareš's work [14] in `BarrettHand/2020-MichalMareš/neural/`.

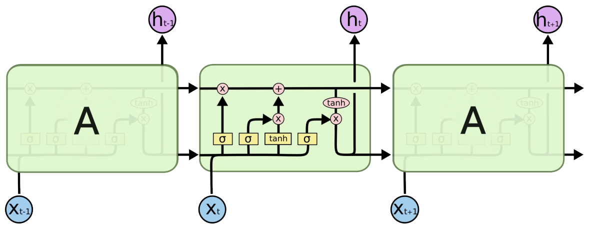


Figure 2.13: Architecture pf LSTM cell. Yellow rectangles are learnable layers, pink circles are pointwise operations, lines merging represent concatenation, forking copies the data into two vectors. This cells are stacked horizontaly and usually also verticaly (rows of cellls are callled layers). Citation from Mareš [7]. Picture from [36].

Data are loaded with specified batch sizes and are padded with zeros to the length of the longest measurement. Then, the time series is passed into the LSTM layer. According to the original length, the output of the last LSTM cell (representing the LSTM-computed features) with nonzero input is selected. The features are then passed into two linear layers to compute the output. Object's category is selected using the softmax layer. During training, the PyTorch automatic differentiation engine is called and used to backpropagate from the loss of output, and the weights are updated according to the set optimizer. This information is taken from Mareš's work [7].

When using LSTM, some parameters have to be set:

1. Number of layers
2. Size of hidden dimension
3. Learning rate
4. Batch-size

As I need to train approximately thirty networks (one for each dataset and ablation experiments), hand-tuning is not an option. I rather used grid search once again. I tried:

1. Two layers: 2, 4;
2. Size of hidden dimension: 32, 64, 128, and 256;
3. Learning rate: logarithmic distribution over $[10^{-5}, 10^{-3}]$;
4. Batch-size: memory of GPU is limit, I used highest possible (some training sets can be surprisingly only one batch)

Therefore, each LSTM training consists of training 336 LSTM models. This usually takes three to four days on GPU grid. The total number of trained networks is around 10 000. The final model was then picked by the best accuracy on the validation set (if there were multiple models with the same accuracy, the one with the most stable learning curve was used). Complete grid search results can be seen on my drive [37] in the folder `*gripper*/Classification/LSTM/Gridsearch/`.

2.3.2 Unsupervised analysis

Principal component analysis

Principal Component Analysis (PCA) is a dimensionality reduction method that is often used to reduce the dimensionality by transforming a large set of variables into a smaller set by preserving as much variance as possible. This allows us to plot vectors from the feature space in a plane and gain some insight into the structure of the data.

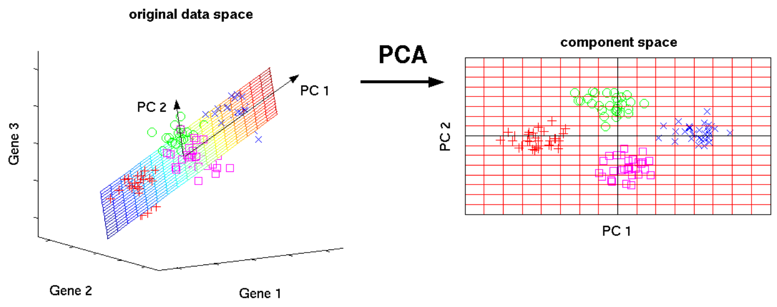


Figure 2.14: Visualisation of the PCA technique. On the left image, a plane define by the first two principal components is visualized in the original space. On the right image, original data are projected into the plane. Picture from [38].

By computing the covariance matrix, finding eigenvectors, and sorting them with the eigenvalue as the key, we can find a new basis with axes that are uncorrelated and ordered by variance. This allows us to visualize high-dimensional data. A proportion of variance represented by each principal component can be computed from eigenvalues, showing how reliable clusters are visible in a plane. I used implementation of SVM from popular python data science package Scikit-learn [39]. My code with dataset loading and features extraction, grid search, and plotting all results can be viewed here [14] in `*robot*/*gripper*/Unsupervised~analysis/PCA/`.

Chapter 3

Data collection, preprocessing and dataset creation

In this section, I will explain how I collected, preprocessed, and split all data used later for classification and in unsupervised analysis.

3.1 Data collection

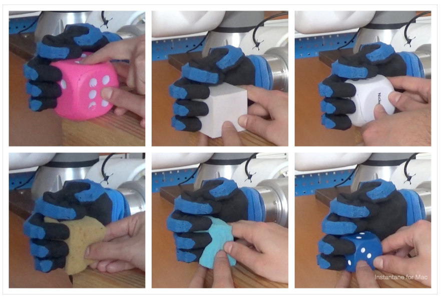
3.1.1 qb SoftHand

I already described the way to control qb SoftHand in Section 2.2.1. The goal was to reach at least 60 samples per object—one sample being one object compression (squeeze) or compression and decompression (squeeze and release)—with the intention of always measuring in the same configuration and preserving some variance. Therefore, measuring in certain configurations was split into series having twenty samples per object each. I managed to do five series on each configuration, attacking 100 samples per object.

It is important to note that we found qb SoftHand was not robust enough for this type of data collection – it had to be repaired four times. The main issue was tearing of the string, which moves all fingers. Moreover, there is some problem with the position encoder, which leads to losing the reference position from time to time. In total, I have done measuring of 6000 samples. However, only 3200 can be considered valid.

Action primitives

No hand parameters can be changed, but we can choose where we place the object. I supposed two actions to maximize and minimize contact with the thumb. The motivation was to study its impact on classification, as thumb opposition is important for primates. I will refer to them as *action 1* (*a1*) and *action 2* (*a2*). In total, we are getting four configurations. Actions can be seen in Fig. 3.1. Later during processing, I split the actions also by considering only “squeeze” and “squeeze and release” part of the recorded time series.


(a) : *action 1* - minimizing contact with thumb.

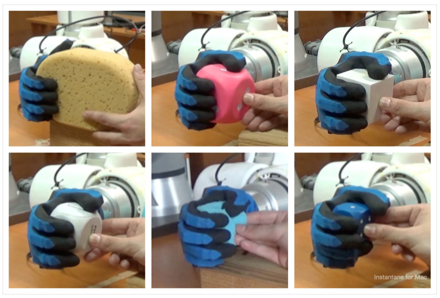

(b) : *action 2* - maximizing contact with thumb.

Figure 3.1: qb SoftHand – Action configurations.

I used two velocities inducted by the waypoints setting:

1. *s1* - waiting 1s; closing 1.5s; waiting 0.5s; closing 1.5s
2. *s2* - waiting 1s; closing 2.5s; waiting 0.5s; closing 2.5s

I will refer them as *s1* and *s2*.

Overview

Roughly 100 samples per object were acquired. Measurements were done only on the ordinary object set (see Fig. 2.2). Two actions and two velocities give four configurations in total. Data was later also processed to “squeeze” and “squeeze and release” parts. Therefore, we will have eight basic datasets. Every measurement has also been video-taped. All my measurements and videos can be found here [40].

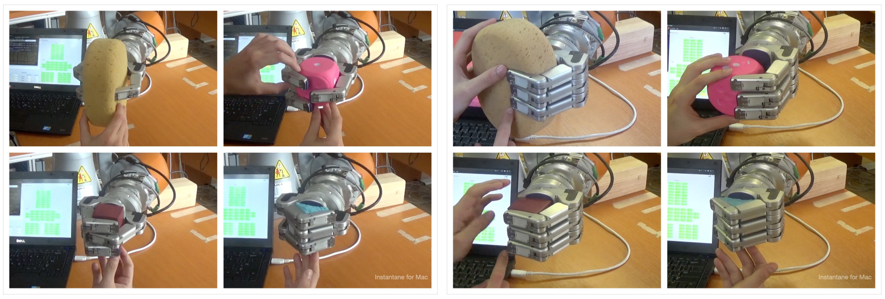
3.1.2 Barrett Hand

Similarly as with the qb SoftHand, the goal was to get at least 60 samples per object—one sample being one object compression (squeeze). Putting the device into operation took more time than expected, so I did not aim to reach a higher number. In total, I have collected 3500 samples. However, only 2500 can be considered valid.

Action primitives

There is a much bigger degree of freedom in setting actions than with qb SoftHand, thanks to two fingers rotating around the base, each mirroring another. I used the following configurations (the same as Michal Mareš used in [7]; see Fig. 3.2):

1. *action 1* — opposite finger configuration (refered as *a1*), Fig. 3.1a
2. *action 3* — lateral finger configuration (refered as *a3*), Fig. 3.1b



(a) : *action 1* – opposing fingers **(b) :** *action 3* – lateral fingers

Figure 3.2: Barrett Hand – action configurations used.

Action 2 also exists, but has not been used.

And similarly as before, two velocities have been chosen:

1. *v0.6* - all three fingers' motors turn at speed 0.6 rad/s
2. *v1.2* - all three fingers' motors turn at speed 1.2 rad/s

I will refer to them as *v0.6* and *v1.2*.

Overview

Roughly 65 samples per object were collected. Measurements were done only on the ordinary objects set. Two actions and two velocities give four configurations in total. Therefore, we will have eight basic datasets. Every measurement has also been recorded. All my measurements and videos can be found here [41].

3.1.3 OnRobot RG6

I did not perform any measurements with this gripper. Instead, I used the measured data available here [22].

Action primitives

Since the gripper is commanded only by the desired force, which is increased incrementally, there is no possibility to set the speed. With a two-finger gripper and highly symmetrical objects (cubes), no other gripping configurations are available.

Overview

Roughly 36 samples per object were collected. Measurements were done only on the ordinary objects set. Speed can not be changed. Therefore, we will have only one configuration. Already measured data was used [22].

3.1.4 Robotiq 2F-85

I did not perform any measurements with this gripper. Instead, I used already measured data [26].

Action primitives

With a two-finger gripper and highly symmetrical objects (cubes), no other gripping configurations are available. Closing/opening speed can be commanded. Four different nominal squeezing velocities were executed:

1. 0.68% (1.6 mm/s)
2. 14.45% (30 mm/s)
3. 50.85% (80 mm/s)
4. 100% (131.33 mm/s)

I will refer to them as *0068*, *1445*, *5085* and *10000*.

Overview

Roughly 60 samples per object were collected. Measurements were done on the ordinary objects set (Fig. 2.1), foams set (Fig. 2.3), and foams subset – Tab. 2.3. Four velocities give four configurations in total. Velocity 0068 was used during measurements on objects set and foams set. All velocities have been used during measurements on the foams subset. Therefore, we will have six basic datasets. Already measured data was used [26].

3.2 Data processing

In the beginning, let us define some useful terms:

1. *Sample* means one squeeze and release of object. On the Barrett Hand, sample is only one squeeze, since recording of releasing was not possible. This was not a problem on other three grippers.

2. *Measurement* means series of samples—squeeze (and release) cycles—usually 20 – 30. On the Barrett Hand, data were collected in individual object compressions, so every measurement was one sample.

3.2.1 qb Softhand, OnRobot RG6 and Robotiq 2F-85

For all of these three grippers, the same processing pipeline was used. Measured time-series (measurements) have two channels and contain 20 samples. The first task is to normalize data. Each channel was scaled by the inverse value of global maxima. The second necessary step is to split the time series in individual samples.

Some samples have errors and need to be discarded. I wanted to do this as precisely as possible, so I decided not to rely only on the automatic splitting triggered, for example, by a position event, and instead wrote a program with a graphical user interface allowing to check each supposed sample and in case of errors marked it as wrong. My first attempt was splitting the signal only by the number of samples, sample length, and starting position – see Fig. 3.3.

Parsing time series by hand

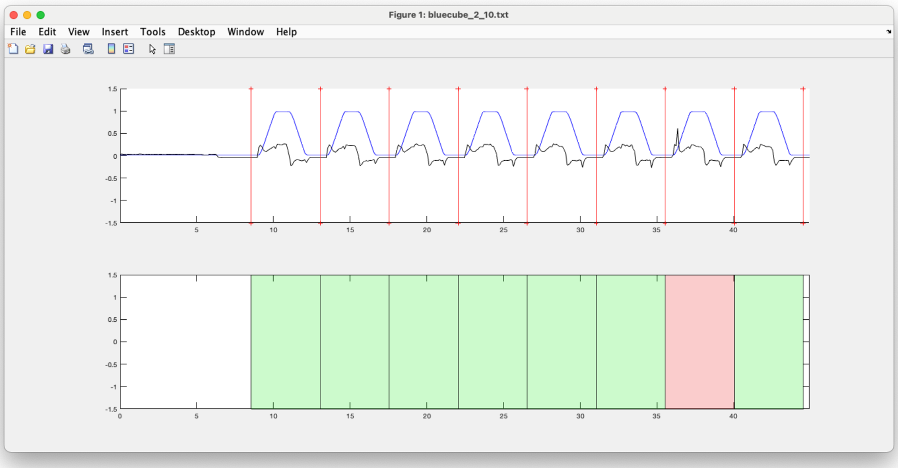


Figure 3.3: First attempt to split measurements. Red rectangles shows discarted measurements.

The upper subplot in Fig. 3.3 shows the recorded time series. The user can set a start just by clicking on a mouse and then use arrows to fine-tune a starting position or increase the number of samples. By pressing $F1$ or $F2$, the sample length can be shortened or increased. Problematic samples can be marked on the second subplot. This can be changed by clicking. When everything is done, by pressing *enter* all samples are split, saved, and a new measurement is loaded. This approach works fine to later feed the LSTM classifier. However, the starting position is not consistent over all samples in relation to, for example, the rising edge of the position signal, which can be problematic when using classification algorithms such as kNN (on the raw data). Therefore, I searched for a more precise way.

Parsing time series using cross-correlation

I noticed that qb SoftHand's position regulation is very precise and tried to take advantage of it. By choosing one representative sample of the position signal and using matlab functions *xcorr* (cross-correlation) and *findpeaks* (maximal position), all samples can be synchronized and cut at the same positions. A new program (see Fig. 3.4) loads measurements in the same way as the first version and saves them by pressing *enter*. Indexes for cutting can be set in the smaller right subplot, and not error-free samples can be once again marked by clicking with a mouse.

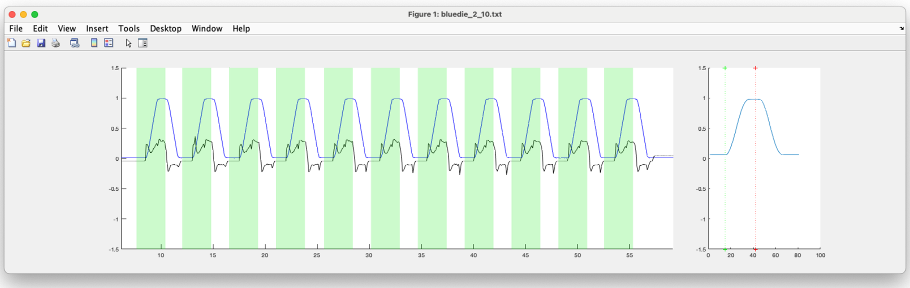


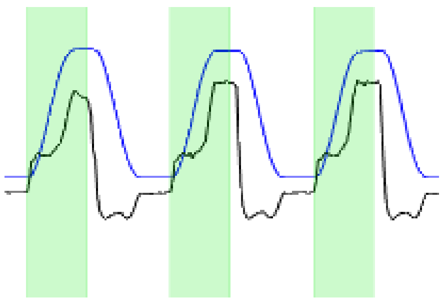
Figure 1: bluedie_2_10.txt
File Edit View Insert Tools Desktop Window Help
1.5
1
0.5
0
-0.5
-1
-1.5
10
15
20
25
30
35
40
45
50
55
0
20
40
60
80
100

Figure 3.4: Final solution. Green rectangles show selected parts on left subplot. This can be set in right subplot. Samples marked as red will be not saved. By pressing *enter*, all valid samples will be saved and new measurement loaded.

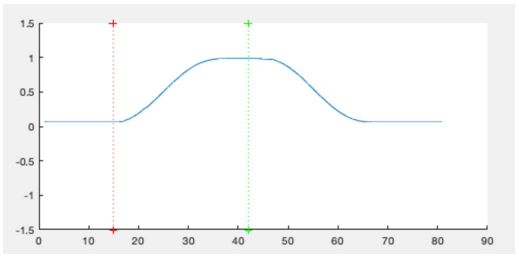
The figure 3.5 illustrates precision of the procedure.

For OnRobot RG6 and Robotiq 2F-85, similarly chosen representative parts of a signal were used. Furthermore, with these grippers, it was impossible to have all samples of the same length; therefore, some of them were artificially enlarged by repeating the starting value. All code is here [14] in `*robot*/*gripper*/Processing/`.

When processing is done, the samples are shuffled (this is an important step) and split into train, validation and test sets.



(a) : Green rectangles show the used part of the sample.



(b) : Two lines can show indexes for cutting. Line can be changed by mouse clicking or pressing backspace; position can be set with use of arrows.

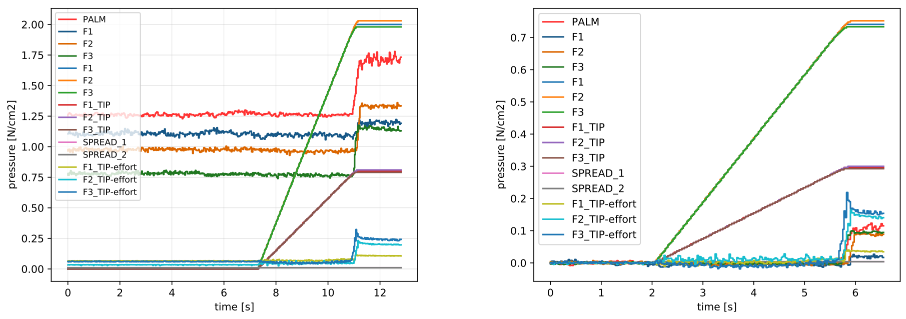
Figure 3.5: Splitting signals into samples with the use of this procedure is precise. Compare green parts of recorded measurement with selected parts of the signal.

3.2.2 Barrett Hand

The Barrett Hand differs from the other grippers by the number of sensory channels and their characteristics. Since some work on it has been already done by Mareš [7], I had a good starting position to decide which way of collecting data will lead to simple processing. As I mentioned in the subsection about Barrett Hand's control in Section 2.2.2, I added saving of events such as *start of a script*, *start of squeezing*, *hand stopping and its* reason (*maximal position*, *pressure threshold* or *getting stuck*). Processing is based mainly on the use of these events.

All code for preprocessing is in `BarrettHand/2021-MichalPliska/Processing/` at [14].

When all measurements are done, the first step is to run `check_rosbag.py` to check and delete bad measurements. I have been doing this manually since I wanted as much control as possible. Then, `make_dataset.py` is next. All channels are scaled by inversion of the highest possible value of a channel (this is known from the beginning). Also, as the hand runs, it is getting hotter, causing a growing offset on tactile sensors. All feature-based methods relying on features such as Minimum value, Maximum value, Mean, etc. can be affected by this. Therefore, for each channel, the average from last one hundred values before *start of squeezing* event is calculated and subtracted. Measurements are already samples by themselves (every compression cycle of the object was collected as a separate measurement). However, measurements have to be the same length in order to use kNN. Before collecting measurements used for dataset generation, I measured few (2–4) pilot measurements on each object and calculated minimal waiting time needed before every squeeze to achieve the same length for all measurements. Thanks to this, measurements can be aligned from the end and cut to the length of maximal global squeezing time (this is simply the time between events *start of squeezing* and *hand stopping*). Samples before and after processing can be seen in Fig. 3.6. Files are saved in .npz format. The last script `split_dataset.py` divided the data into train, validation and testing sets with specified amounts.



(a) : Unprocessed signal. Notice the offset of tactile data.

(b) : Processed signal. Rest value is now zero for all channels.

Figure 3.6: Measurement/Sample (same on Barrett Hand) before and after processing.

3.3 Generated datasets

In this section, I briefly present all datasets generated from measurements later used for classification and unsupervised analysis.

3.3.1 qb Softhand

Per each object, 20 samples are in the test set, 20 in the validation set, and the rest (60+) in the train set. With 9 objects in the dataset, this makes in total 180 samples in validation and testing set and the rest for training (between 500 and 600 samples).

Dataset	Set	Action	Velocity	Train	Validation	Test
a1s1-s	objects	a1	s1	573	180	180
a1s2-s	objects	a1	s2	604	180	180
a2s1-s	objects	a2	s1	578	180	180
a2s2-s	objects	a2	s2	577	180	180
a1s1-sq	objects	a1	s1	583	180	180
a1s2-sq	objects	a1	s2	576	180	180
a2s1-sq	objects	a2	s1	577	180	180
a2s2-sq	objects	a2	s2	539	180	180
squeeze	objects	all	all	1894	600	600
squeeze and release	objects	all	all	1849	600	600

Table 3.1: qb SoftHand datasets.

Shorter "s" denotes datasets using only the "squeeze" part of the samples, the longer "sq" denotes using of "squeeze and release" part. Dataset labels in italics denote datasets formed by merging several individual datasets (here *squeeze* and *squeeze and release*). Notice that validation and test sets in *squeeze* and *squeeze and release* do not have 720 samples, but only 600. That is because for objects *yellowsponge*, *bluedie* and *ycbcube*, there is no difference between actions and samples measured as the *action 1* was also used as the *action 2*. This means that part of *a1* and *a2* datasets is shared and when they are merged, overlap occurs. No dataset should contain proportionally fewer measurements in order to avoid getting better results by solving an easier problem.

3.3.2 Barrett Hand

Per each object, 15 samples are in the test set, 15 in the validation set and the rest (35+) in the train set. With 9 objects in the dataset, this makes in total 135 samples in validation and testing set and the rest for training (over 300 samples).

Dataset	Set	Action	Velocity	Train	Validation	Test
a1v0.6	objects	a1	v0.6	333	135	135
a1v1.2	objects	a1	v1.2	346	135	135
a3v0.6	objects	a3	v0.6	341	135	135
a3v1.2	objects	a3	v1.2	346	135	135
a1	objects	a1	all	679	270	270
a3	objects	a3	all	687	270	270
all	objects	all	all	1336	540	540

Table 3.2: Barrett Hand datasets.

3.3.3 OnRobot RG6

Per each object, 6 samples are in the test set, 6 in the validation set and the rest (±22) in the train set. This dataset is generally too small for robust

Dataset	Set	Action	Velocity	Train	Validation	Test
objects	objects			198	54	54

Table 3.3: OnRobot RG6 datasets.

results. However, in this case, the classification was so accurate that it is not a problem.

3.3.4 Robotiq 2F-85

Per each object, 15 samples are in the test set, 15 in the validation set and the rest (±28) in the train set.

Dataset	Set	Action	Velocity	Train	Validation	Test
objects	objects		0.68%	244	135	135
foams	foams		0.68%	502	300	300
foams-smaller-0068	foams subset		0.68%	145	90	90
foams-smaller-1445	foams subset		14.45%	158	90	90
foams-smaller-5085	foams subset		50.85%	143	90	90
foams-smaller-10000	foams subset		100.00%	167	90	90

Table 3.4: Robotiq 2F-85 datasets.

Chapter 4

Results

The first part of this chapter deals with classification. I used four classifiers described back in subsection 2.3.1 (kNN with time series, kNN with features, SVM with features and LSTM with time series). After evaluating each gripper separately (data from only one gripper), the results are compared together to make conclusions about the gripper's utility. The last two sections then speak about knowledge transfer (how well the models perform on datasets that they were not trained on) and the ablation effect (ablation in the meaning of suspending some channels, like position or current, for example). The classifier used for the last two tasks is LSTM because we can expect the greatest generalisation from it.

In the second part, PCA is done on selected datasets, looking further to find patterns in the data.

4.1 Classification

All results presented are performance on the test sets. I used a heatmap and not a classic table for visualizing the results. Although the labels may be harder to read, I think it gives better insight. All classifiers' hyperparameters and training parameters can be found in Appendix A.

4.1.1 qb Softhand

Here, "squeeze" and "squeeze and release" datasets can not be evaluated with kNN, as they contain samples of various lengths from different actions. Best results are delivered by LSTM. It is also interesting that such a simple algorithm as kNN performs similarly. On the other hand, feature-based classification is the most unreliable one (see Fig. 4.1).

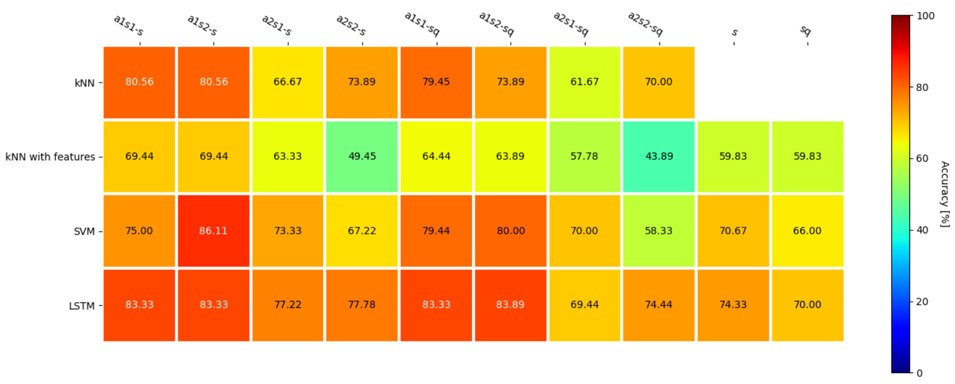


Figure 4.1: qb Softhand – Overview of classification performance over datasets. Performance of different classifiers is in rows, with increasing complexity of the classifier from top to bottom. Different datasets used are in columns. Please note, that "s" means "squeeze" and "sq" means "squeeze and release" datasets.

Confusion matrices illustrate the pattern of misclassification—one representative is shown in Fig. 4.2. Note how misclassification correlates with the distance between objects in the material elasticity dimension in Fig. 2.2.

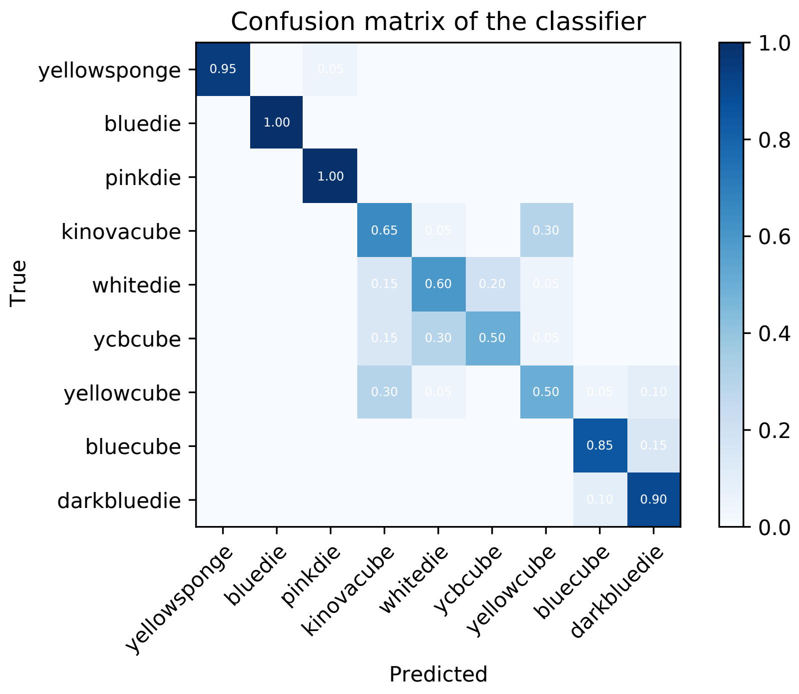


Figure 4.2: qb Soft Hand – confusion matrix. The model used here as a representative is LSTM a2s1-a.

4.1.2 Barrett Hand

All classifiers perform stunningly well on the Barrett Hand (see Fig. 4.3). In fact, kNN performs better than LSTM. This probably means that the data are straightforward and there is not too much hidden information. It agrees with my observation that samples can be classified just by visual inspection. Confusion matrices are not shown here due to the almost perfect performance of classification.

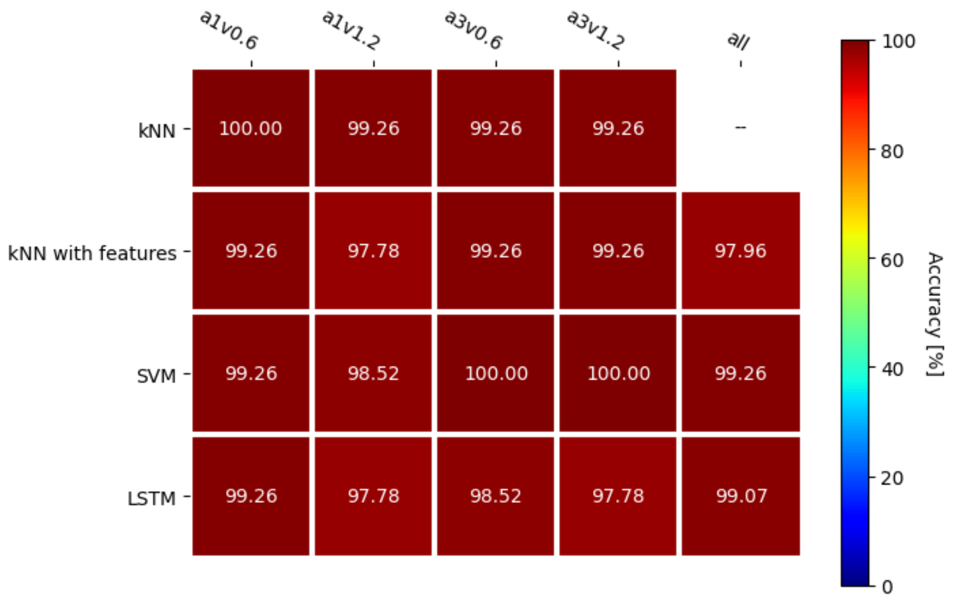


Figure 4.3: Barrett Hand – Overview of classification performance over datasets. Performance of different classifiers is in rows, with increasing complexity of the classifier from top to bottom. Different datasets used are in columns.

4.1.3 OnRobot RG6

Again, all classifiers perform really well with a slump on kNN features (see Fig. 4.4). However, almost 90% is still high accuracy.

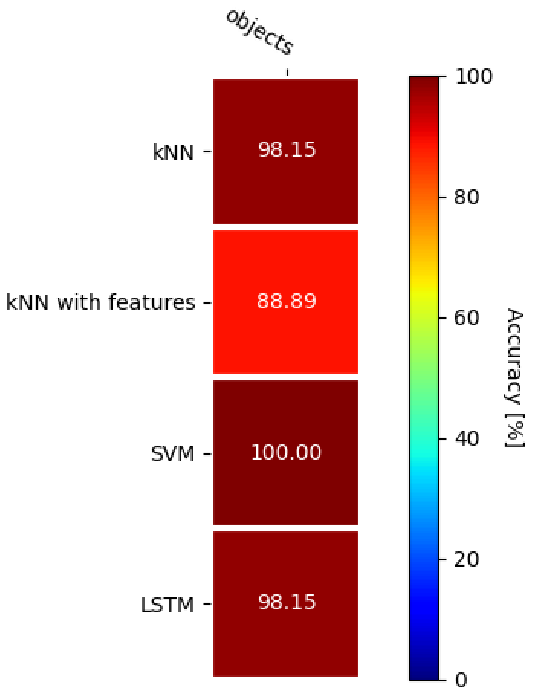


Figure 4.4: OnRobot RG6 – Overview of classification performance over datasets. Performance of different classifiers is in rows, with increasing complexity of the classifier from top to bottom. Different datasets used are in columns.

4.1.4 Robotiq 2F-85

Performance on 0.68% of maximal velocity on all object sets looks similar to the Barrett Hand and OnRobot RG6. However, it drops with higher speeds and simple classifiers, and we can finally see the expected power of LSTM (see Fig. 4.5). Interestingly, SVM performs the same. This trend can be seen across all grippers. Because the training of LSTM with grid search takes two days and SVM only ten minutes, it should be highlighted.

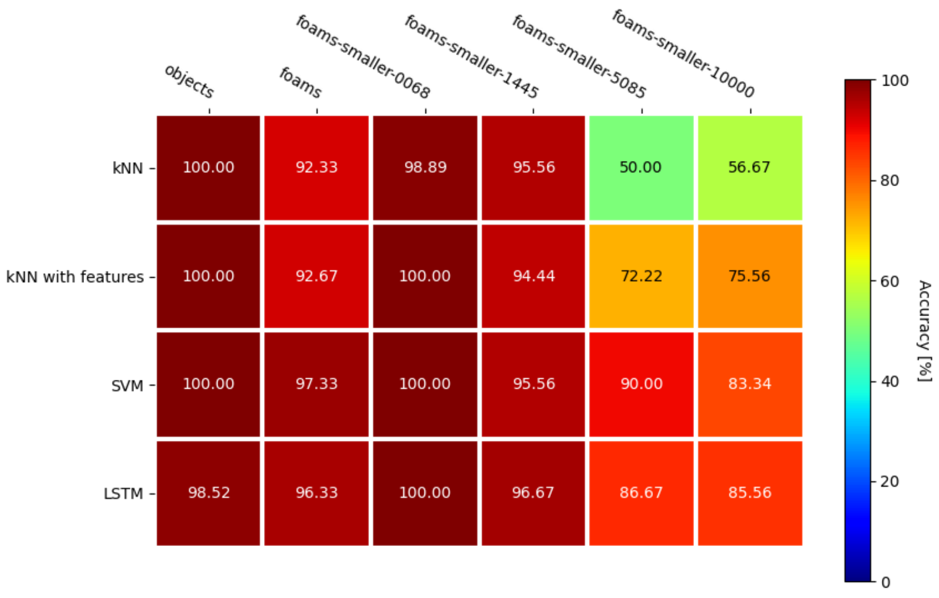


Figure 4.5: Robotiq 2F-85 – Overview of classification performance over datasets. In the left column, classifiers are ordered from the simplest to the most sophisticated ones. Different datasets used are in columns.

Best performance is achieved with velocity 0.68%, but 14.45% also works well. Surprisingly, not only objects but also foams can be classified well – see Fig. 4.6. Misclassification does not seem significant.

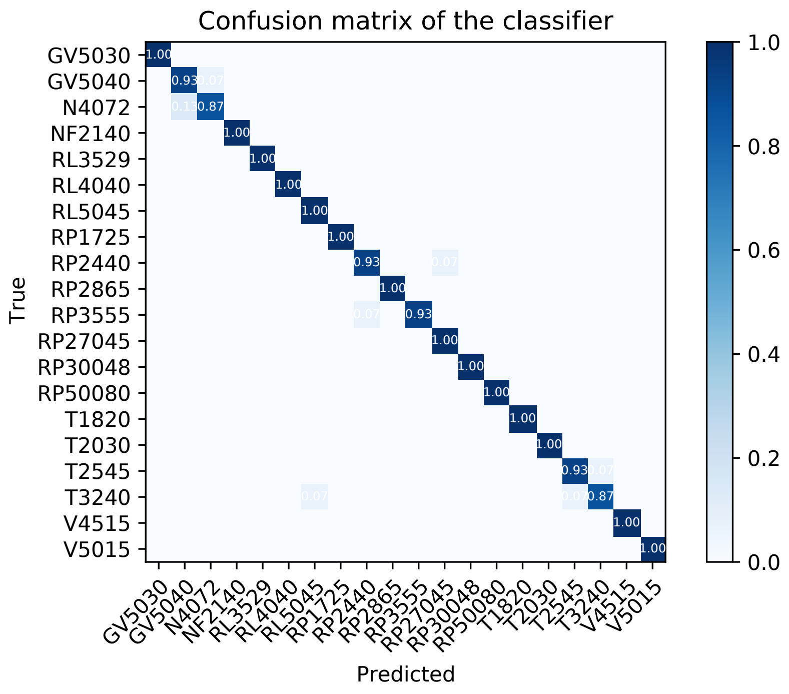


Figure 4.6: Robotiq 2F-85 – confusion matrix – Foams classification. Squeezing with 0.68% of maximal velocity. The model used here is SVM foams.

4.1.5 Gripper comparison

Now, let us compare all grippers together (see Fig. 4.7). For qb SoftHand, I pick "squeeze' and "squeeze and release" datasets as representatives. However, kNN can not be run on those. Therefore, the average of individual actions is used to get at least some idea. Then for the Barrett Hand, I used the "all" dataset. I tried to isolate the influence of individual actions, as they do not exist on the last two grippers. With OnRobot RG6 and Robotiq 2F-85, only the "objects" datasets are available as an option.

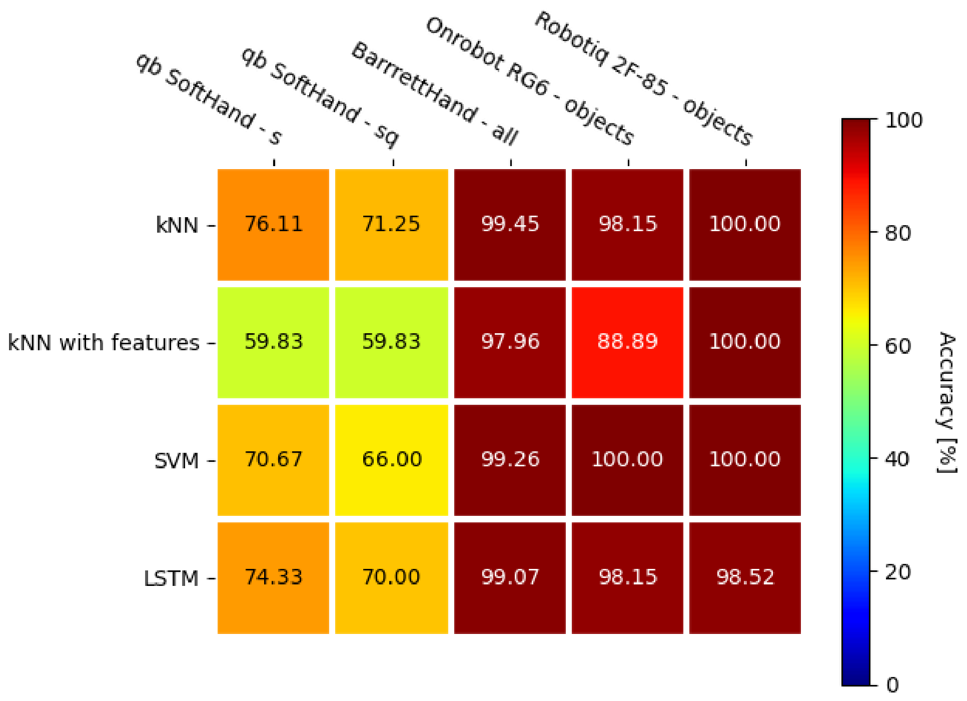


Figure 4.7: All grippers – Overview of classification performance over datasets. Performance of different classifiers is in rows, with increasing complexity of the classifier from top to bottom. Different datasets used are in columns.

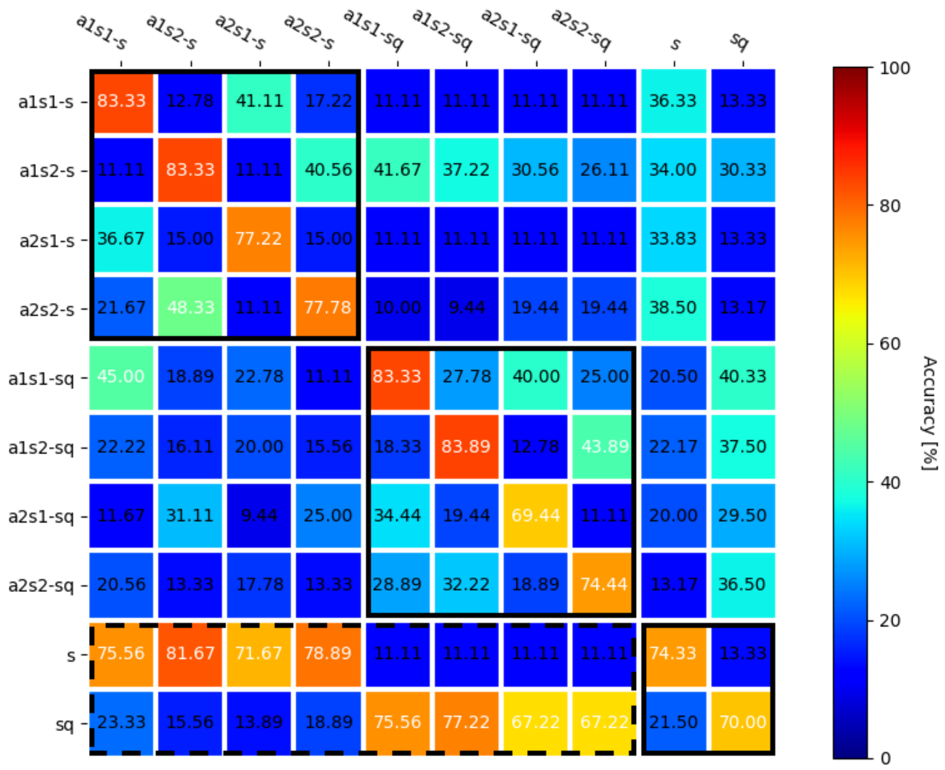
We can see that the worst performances are delivered by qb SoftHand. Surprisingly, it looks that the classification works better on samples only with the "squeeze" part, even though we hypothesized that the "release" part could add additional information. The best grippers are probably Barrett Hand and Robotiq 2F-85 (showing that it can also correctly classify foams). Note that Robotiq 2F-85 has only two channels, unlike Barrett Hand with 107. OnRobot RG6 also works well. However, data may be more complex because we can see a slump when using only features.

4.1.6 Knowledge transfer

When all models are trained, we can test how well each of them generalizes to other datasets. In addition, to bring another insight to the structure of the data, it may also reveal if training on some actions and velocity is more effective. It is worth mentioning that due to the fact of having 9 objects in the objects set, the random classifier accuracy should be 11.11%. All experiments are done with LSTM classifiers.

qb SoftHand

For better orientation, I marked the areas of transfer on the “squeeze” domain (big left upper frame), “squeeze and release” domain (big middle frame), “merged datasets” domain (small right lower frame), and “merged datasets decomposition” domain (left lower dashed frame). The diagonal shows the accuracy on the datasets used for training.



	a1s1-s	a1s2-s	a2s1-s	a2s2-s	a1s1-sq	a1s2-sq	a2s1-sq	a2s2-sq	s	sq
a1s1-s	83.33	12.78	41.11	17.22	11.11	11.11	11.11	11.11	36.33	13.33
a1s2-s	11.11	83.33	11.11	40.56	41.67	37.22	30.56	26.11	34.00	30.33
a2s1-s	36.67	15.00	77.22	15.00	11.11	11.11	11.11	11.11	33.83	13.33
a2s2-s	21.67	48.33	11.11	77.78	10.00	9.44	19.44	19.44	38.50	13.17
a1s1-sq	45.00	18.89	22.78	11.11	83.33	27.78	40.00	25.00	20.50	40.33
a1s2-sq	22.22	16.11	20.00	15.56	18.33	83.89	12.78	43.89	22.17	37.50
a2s1-sq	11.67	31.11	9.44	25.00	34.44	19.44	69.44	11.11	20.00	29.50
a2s2-sq	20.56	13.33	17.78	13.33	28.89	32.22	18.89	74.44	13.17	36.50
s	75.56	81.67	71.67	78.89	11.11	11.11	11.11	11.11	74.33	13.33
sq	23.33	15.56	13.89	18.89	75.56	77.22	67.22	67.22	21.50	70.00

Figure 4.8: qb SoftHand – Knowledge transfer. In rows, classifiers are ordered according to the datasets they were trained on. Columns are labeled with the dataset on which they were tested.

First, the knowledge transfer between the “squeeze” and “squeeze and release” domains is poor (see Fig. 4.8). High accuracy on the “s” type dataset does not correlate with accuracy on the “sq” type dataset—otherwise, we would see parallel diagonals in the blue areas. Moreover, the transfer over action does not perform very well (follow 2x2 squares on the main diagonal). However, we can see some correct classification over speeds, but this is expected as 33.3% of a1 and a2 datasets is shared (this is mentioned in Chapter 3—there is no difference between actions on three of all nine objects). Therefore, I suggest that it is more about the use of the same samples than about actual knowledge transfer. In summary, it seems that the primary source of variability is the “squeeze” vs. “squeeze and release” type of data, followed by squeezing speed and then action type. This hypothesis will be verified later using PCA analysis.

The right upper strip shows what is now expected. Models trained on the basic dataset can achieve only partial accuracy on the merged datasets. Even on the merged datasets, transfer between the “squeeze”’ and “squeeze and release” domains does not occur. The “merged datasets decomposition” domain reveals the accuracy on the basic datasets of models trained on the merged ones. Accuracy correlates with accuracy achieved with training on basic sets less than I would expect.

Barrett Hand

With the Barrett Hand, generalization between velocities works surprisingly well. However, there is little transfer between actions (see Fig. 4.9)—the right upper strip reflects that. This is completely opposite from what we have seen for qb SoftHand and constitutes again a hypothesis to be tested in unsupervised analysis—principle clusters should form based on finger configuration rather than squeezing speed.

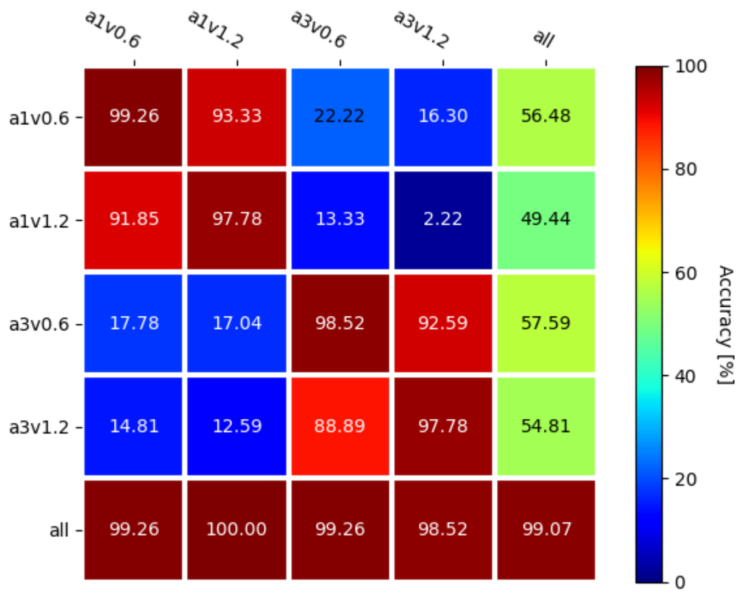
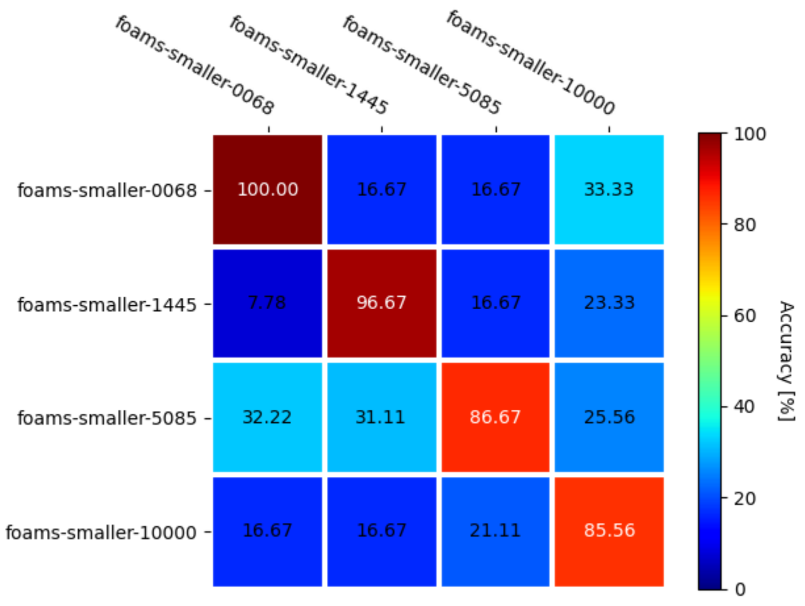


Figure 4.9: Barrett Hand – Knowledge transfer. In rows, classifiers are ordered according to the datasets they were trained on. Columns are labeled with the dataset on which they were tested.

Robotiq 2F-85

Note that this time, models on the foams subset instead of objects set are studied. Any generalization can hardly be found since there is no visible pattern.



	foams-smaller-0068	foams-smaller-1445	foams-smaller-5085	foams-smaller-10000
foams-smaller-0068	100.00	16.67	16.67	33.33
foams-smaller-1445	7.78	96.67	16.67	23.33
foams-smaller-5085	32.22	31.11	86.67	25.56
foams-smaller-10000	16.67	16.67	21.11	85.56

Figure 4.10: Robotiq 2F-85 – Knowledge transfer. In rows, classifiers are ordered according to the datasets they were trained on. Columns are labeled with the dataset on which they were tested.

4.1.7 Ablation study

Since there are multiple sensory channels, we can study their impact on classification and look for the main source of information. All experiments are done with the LSTM classifier.

qb Softhand

From Fig. 4.11 it is obvious that the primary source of information is the motor current channel. It is interesting that when using only this channel, accuracy is better than when using both current and position.

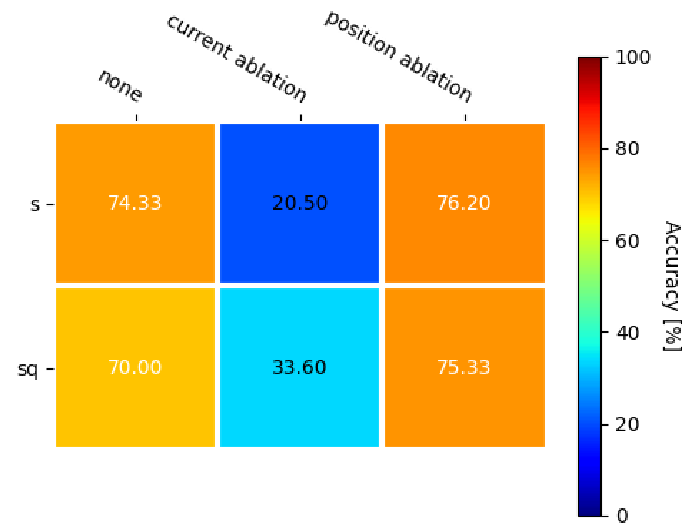


Figure 4.11: qb Soft Hand – Ablation. Datasets are in rows. Columns are labeled based on which sensory channel was suspended.

Barrett Hand

It is good to point out that 96 of the 107 Barrett Hand channels are tactile sensors (pressure-sensitive). However, even without tactile data, the classification is nearly 100% (see Fig. 4.12). In addition, let us consider the fact that without tactile channels, the rest of the channels (position and effort) have similar characteristics as the other grippers.

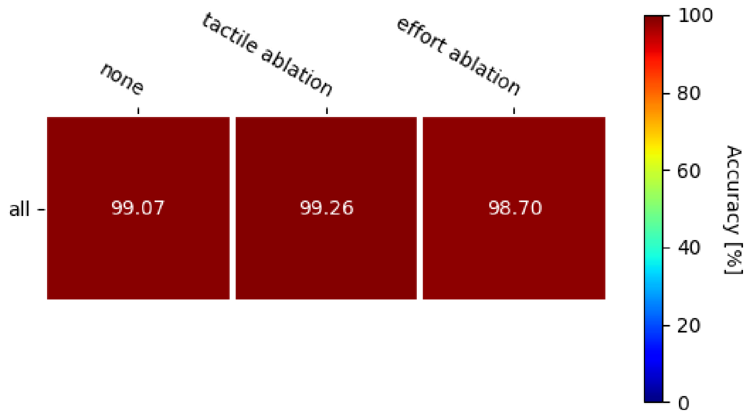


Figure 4.12: Barrett Hand – Ablation. Datasets are in rows. Columns are labeled based on which sensory channel was suspended.

OnRobot RG6

Data can be classified without error by using only width data. Force seems unnecessary. However, it still holds some information—see Fig. 4.13.

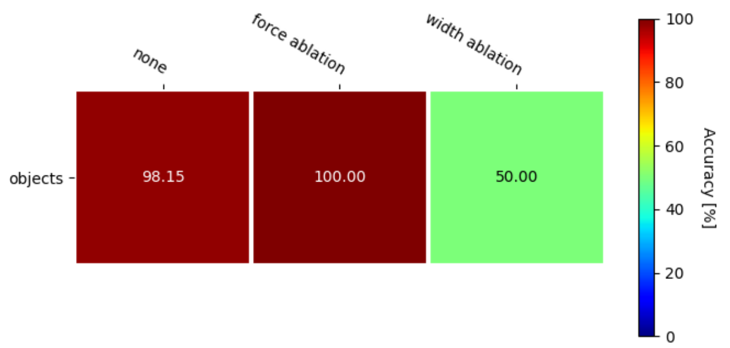


Figure 4.13: OnRobot RG6 – Ablation. Datasets are in rows. Columns are labeled based on which sensory channel was suspended.

Robotiq 2F-85

This gripper already shows high utility when classifying all foams well (see Fig. 4.6. From this experiment can be seen that each channel holds information sufficient for good classification (see Fig. 4.1.7). However, the current signal is a little bit more critical.

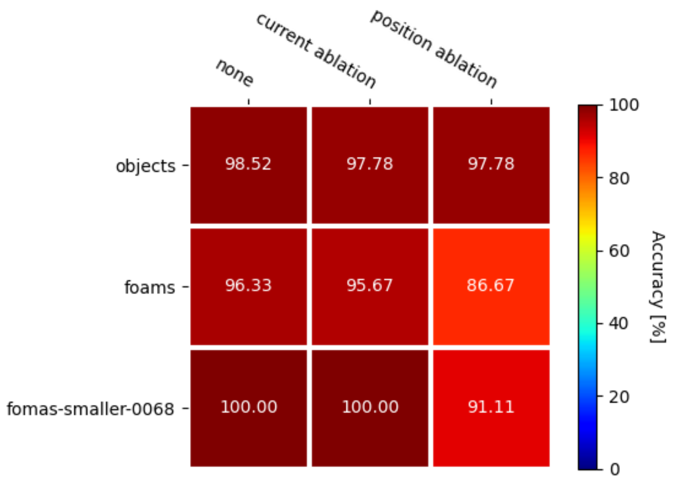
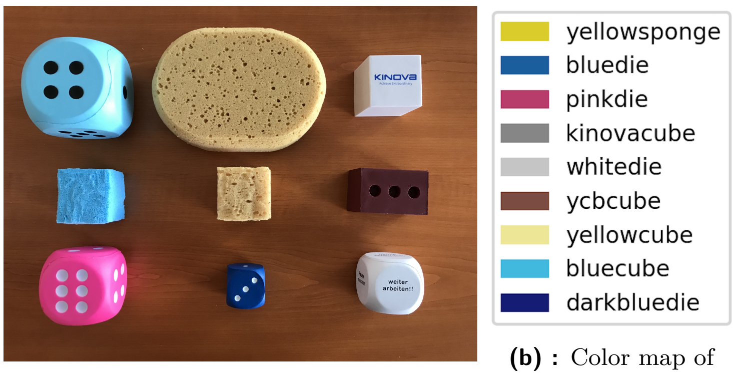


Figure 4.14: Robotiq 2F-85 – Ablation. Datasets are in rows. Columns are labeled based on which sensory channel was suspended.

4.2 Unsupervised analysis

To complement the classification results from the previous section and to get additional insights into the structure of sensory data collected during squeezing soft objects, I used Principal Component Analysis (PCA) to visualize what the main sources of variability in individual datasets are. In particular, is it the properties of the gripper, the action parameters (speed, finger configuration), or the objects being compressed by the gripper?

The color matches the colors of objects. If two objects have the same color, a lighter shade is used for a smaller one (see Fig. 4.15). I suggest looking at Fig. 2.2 before further reading. Each gripper has a unique symbol used for plotting: qb SoftHand uses horizontal triangles, Barrett Hand vertical triangles, OnRobot RG6 is marked with circles, and squares were chosen for samples from Robotiq 2F-85. Three of the four grippers have two sensory channels and therefore is PCA made in twenty-dimensional space (2 time series and 10 features per channel). Barrett Hand has 107 channels; therefore, the number of dimensions is 1070.



(a) : The objects set.

(b) : Color map of the objects.

Figure 4.15: Objects and their color map. The color matches the colors of objects. If two objects have the same color, a lighter shade is used for a smaller one.

4.2.1 qb Softhand

At first, I analyzed all samples together from qb SoftHand. Fig. 4.16 reveals that the main source of variance is "squeeze" or "squeeze and release" attribute (a small black dot is used for differentiation)—see the two principal clusters. It looks like both of them have other subclusters.

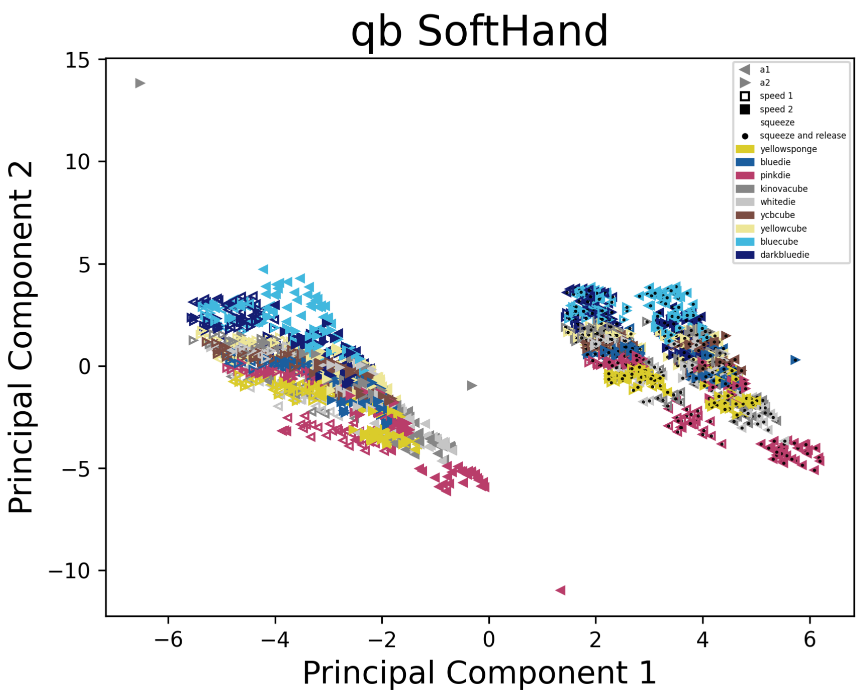
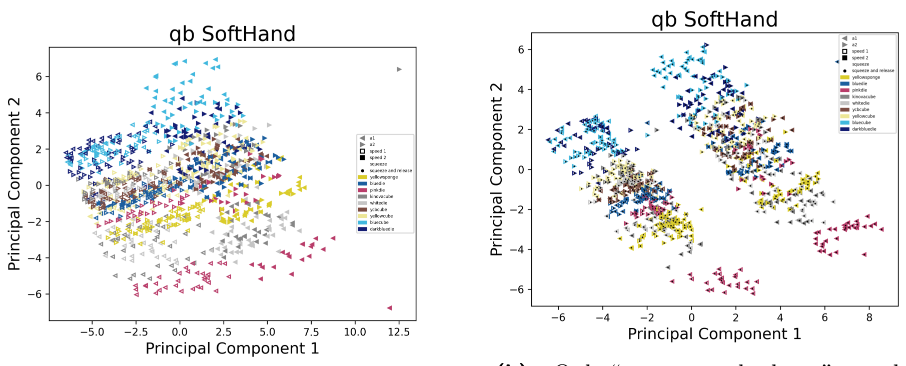


Figure 4.16: qb SoftHand – PCA. All measured samples were analyzed together. The first principal component captures 59% of variance, and the first two then 77%. We can see two clusters: "squeeze" vs. "squeeze and release" samples.

Therefore, I plotted "squeeze" and "squeeze and release" samples separately in Fig. 4.17. Subclusters are formed by velocity change. This agrees with the prediction made in Section 4.1.6.



(a) : Only "squeeze" samples. The first principal component captures 53% of variance, and the first two then 79%.

(b) : Only "squeeze and release" samples. The first principal component captures 45% of variance, and the first two then 74%.

Figure 4.17: qb SoftHand – PCA. A detailed look at two main clusters ("squeeze" vs. "squeeze and release" samples) – their individual subclusters form based on velocity.

I also wanted to take a look only at one action configuration independently. From Fig. 4.18, it seems that clusters are made based on effort during squeezing (go from lower right corner to upper left).

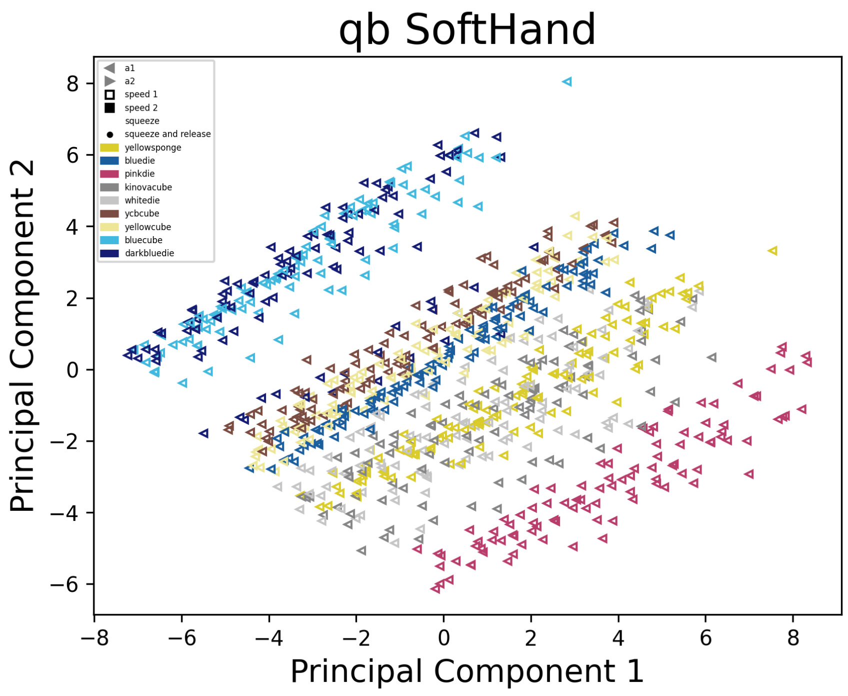
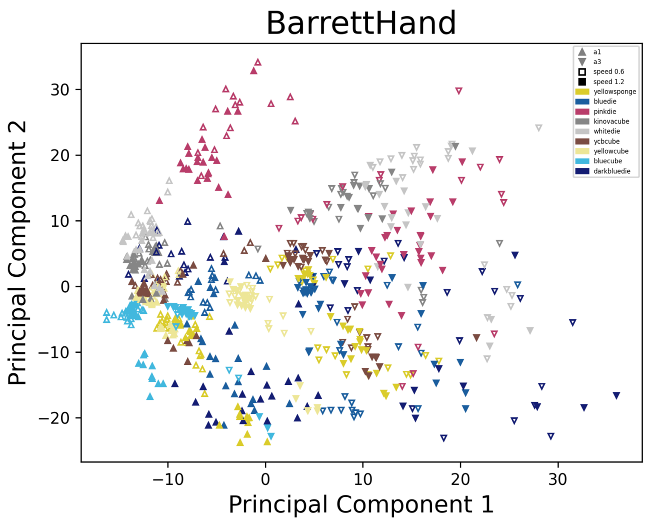


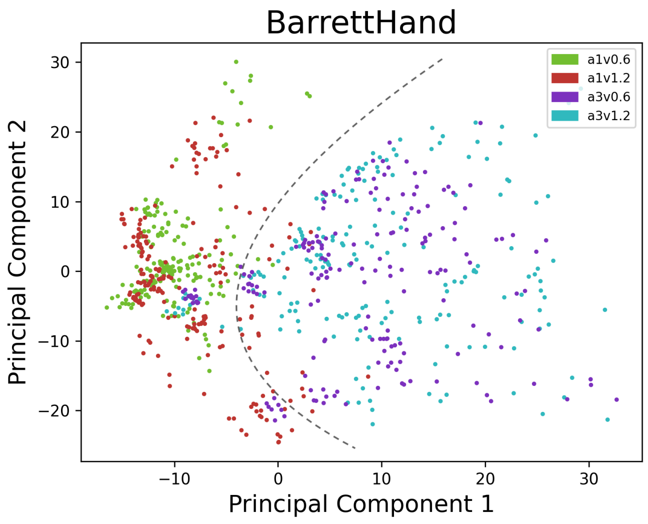
Figure 4.18: qb SoftHand – PCA. a1s1-s samples separately. The first principal component captures 49% of variance, and the first two then 81%. The clusters roughly match with the effort during grasping.

4.2.2 Barrett Hand

The feature space of Barrett Hand has 1070 dimensions. The first subfigure in Fig. 4.19 shows all measured samples together. Some clusters are visible, but it is hard to distinguish. However, notice that the plane can be split into *action 1* and *action 3* regions. To point this out clearly, I made a second subfigure where colors code only actions and drew a separating line. This agrees with the prediction made in Fig. 4.1.6—grasping configuration is indeed the primary source of variance.



(a) : All configurations and objects.



(b) : Only configurations.

Figure 4.19: Barrett Hand – PCA. All samples. The first principal component captures 12% of variance, and the first two then 22%.

I also explored samples from one action only – see Fig. 4.20. I picked *a1v0.6* (lateral finger configuration and lower velocity) samples as representative. We can see clusters forming. Going along the axis of the first quadrant, size seems to be the main factor. Samples above the axis are those measured on harder objects.

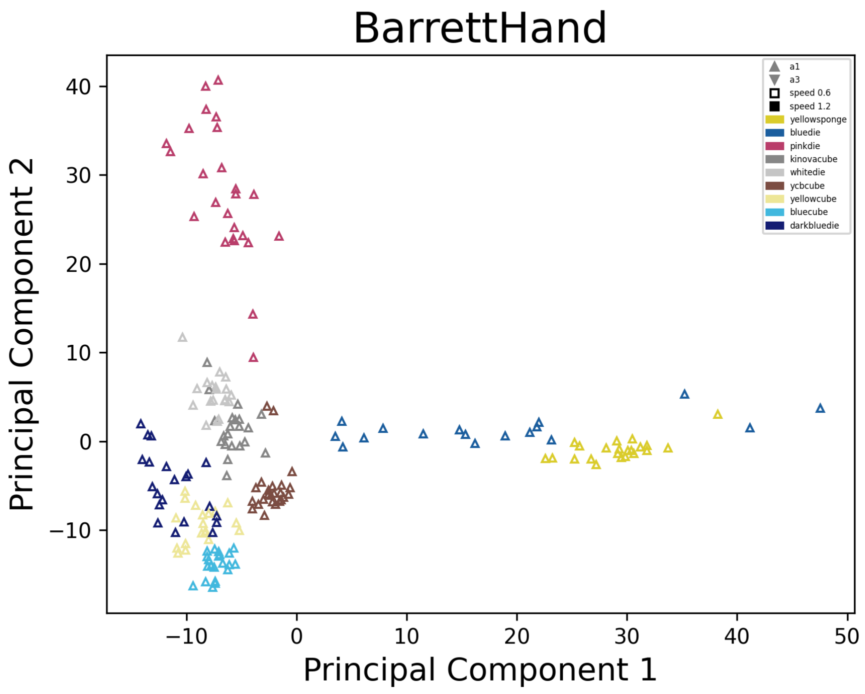


Figure 4.20: Barrett Hand – PCA. a1v0.6 samples separately. The first principal component captures 18% of variance, and the first two then 31%. It seems that the first principle roughly matches size and second stiffness.

4.2.3 OnRobot RG6

Only one dataset was measured on OnRobot RG6. One direction in the plane forms clusters and the second keeps intra-class variance (see Fig. 4.21). Clusters are well pronounced, which is consistent with high classification accuracy. It also seems that the elasticity rather than volume is separating the objects. This was expected for a 2-finger gripper.

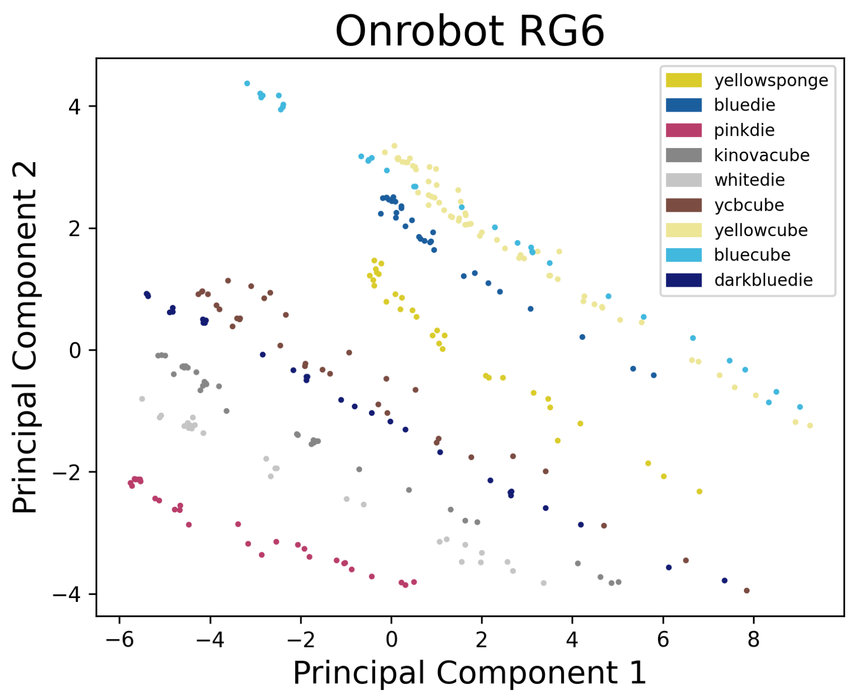


Figure 4.21: OnRobot RG6 – PCA. Objects samples (only one grasping configuration exists). The first principal component captures 60% of variance, and the first two then 81%. The clusters roughly matches with the effort during grasping.

4.2.4 Robotiq 2F-85

Similarly to OnRobot RG6, there is only one dataset measured on ordinary objects to evaluate. This gripper delivers the best pronounced cluster (see Fig. 4.22). This matches with the observations in ablation study (Section 4.1.7), where we could see that information from any channel is sufficient for classification.

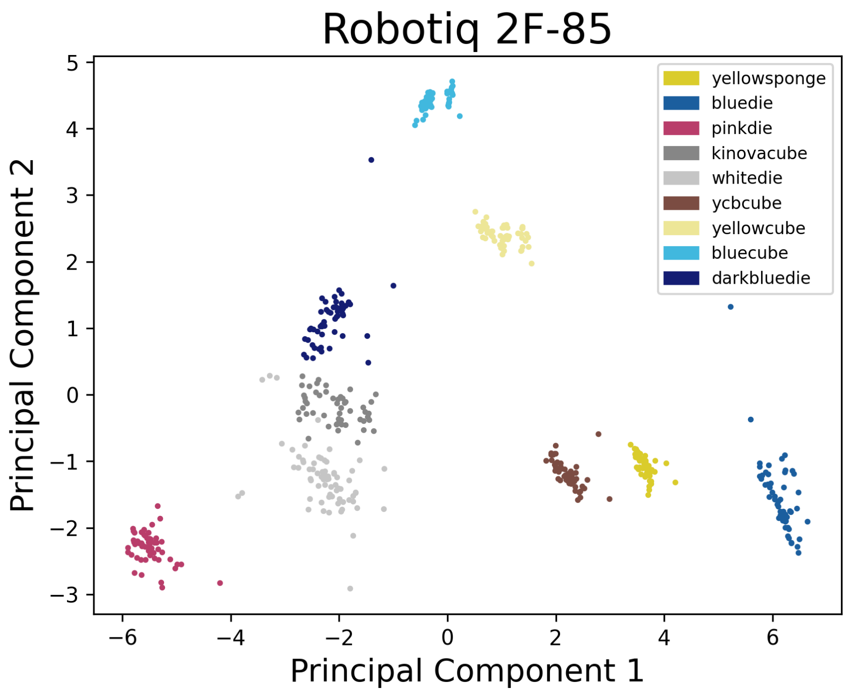


Figure 4.22: Robotiq 2F-85 – PCA. Objects samples (only one grasping configuration exists). The first principal component captures 59% of variance, and the first two then 82%.

Chapter 5

Conclusion, discussion and future work

5.1 Conclusion

In this work, I studied the classification of deformable objects by grasping them using 4 different robot hands/grippers: Barrett hand (3 fingers with adjustable configuration), qb SoftHand (5 fingers, 1 motor), and two industrial type parallel jaw grippers (Robotiq 2F-85 and OnRobot RG6). The time series collected during object compression (and sometimes decompression) were fed into four different classifiers: k Nearest Neighbors (kNN) and LSTM applied on raw data and kNN and SVM on features. I systematically compared the grippers' performance, together with the effects of: (i) action parameters (grasping configuration and speed of squeezing), (ii) knowledge transferability, and (iii) individual sensory modalities.

Three object sets were used: (i) 9 ordinary deformable objects, (ii) 20 polyurethane foams, and (iii) 6 polyurethane foams subset. All hands/grippers reached a good performance on the first set ("ordinary objects"). The Robotiq 2F-85 and the Barrett Hand performed best. The OnRobot RG6 was closely in line, and qb SoftHand performed significantly worse. On the second set (polyurethane foams), only the Robotiq 2F-85 gripper was tested and performed well. Note that this is clearly a superhuman performance on the foams set.

Regarding the performance of different classifiers, SVM performed similarly to LSTM over all hands/grippers. Because the training of SVM takes 10 minutes on a PC and the training of LSTM takes three to four days on GPU a grid, the performance of SVM should be highlighted.

Comparing the individual grippers, the Robotiq 2F-85 and the Barrett Hand performed best. However, the Barrett Hand has 107 sensory channel, while the Robotiq 2F-85 only 2. The OnRobot RG6's performance is closely in line. The qb SoftHand performed worst. The 2-finger grippers thus provide a parsimonious solution to deformable object classification relying only on the stress/strain characteristics in the 2 sensory channels—provided that there is a difference in the stiffness/elasticity of the objects.

The utility of Robotiq 2F-85 has been confirmed by ablation experiments. They clearly showed that only one channel is sufficient for high accuracy classification. The same is true for OnRobot RG6's position channel. The Barrett Hand's performance with sensor's modalities also sustained high. However, the ablation experiment here was not so systematic (training LSTM models on the Barrett Hand's datasets took more time). In addition, the generalization rate was tested by evaluating the model on other than its training set. However, knowledge transfer has proven to work only on the domain of same Barrett Hand's actions.

To complement supervised learning, PCA was done, showing that the structure of the data is not complicated at all and can be explained quite well. The conjectures from the knowledge transfer experiment have been confirmed here. The quality of clusters also strongly correlates with the accuracy of kNN with features. On most of the hands/ grippers, +60% variance was captured by the first two principal components. On the Barrett Hand, it was only ±30%.

5.2 Discussion and future work

In this section, I am going to discuss the limitations of the taken approach and possible future work.

One of the main problems were difficulties with the robot hardware during the extensive data collection. The qb SoftHand broke down several times and had to be repaired (typically by replacing the string pulling all the joints). This may be part of the reasons behind the poorer classification with this device, since the internal hand configuration may have differed between measurements. The Barrett Hand was not sending the sensory feedback for a long time until a cable was replaced. Additionally, much time was spent on attempts to record the release (decompression) part of a single grasp, but I later found out that the internal controller of the hand does not allow such measurements and issues a protective stop.

In the future, several concepts would be worth pursuing, such as:

1. Doing more systematic ablation study on the Barrett Hand
2. Completing measurements on foams set and subset also with other grippers
3. Doing PCA on foams datasets
4. Finding real ground true values of stiffness and density in order to try regression
5. Decrease LSTM models capability by lowering its number of layers and hidden dimensions in order to see hand's/gripper's limitation

6. Try the autoencoder architecture and do PCA of latent space in order to see how model-free learning can be useful when manipulating unseen objects

Bibliography

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, “Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey,” *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018. [Online]. Available: https://journals.sagepub.com/doi/10.1177/0278364918779698

[3] M. V. Liarokapis, B. Calli, A. J. Spiers, and A. M. Dollar, “Unplanned, model-free, single grasp object classification with underactuated hands and force sensors,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 5073–5080. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/7354091

[4] J. Bednarek, M. Bednarek, P. Kicki, and K. Walas, “Robotic touch: Classification of materials for manipulation and walking,” in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019, pp. 527–533. [Online]. Available: https://ieeexplore.ieee.org/document/8722819

[5] M. Hoffmann, K. Štěpánová, and M. Reinstein, “The effect of motor action and different sensory modalities on terrain classification in a quadruped robot running with multiple gaits,” *Robotics and Autonomous Systems*, vol. 62, no. 12, pp. 1790–1798, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092188901400133X

[6] P. Stoudek, “Extracting material properties of objectsfrom haptic exploration using multiplerobotic grippers,” Master’s thesis, 2020. [Online]. Available: https://dspace.cvut.cz/handle/10467/88162

[7] M. Mares, “Exploratory action selection to learn object properties from haptic exploration using a robot hand - bachelor’s thesis,” 2020. [Online]. Available: https://dspace.cvut.cz/handle/10467/87889

[8] “Polymeric materials, cellular flexible – Determination of stress-strain characteristics in compression,” 1986. [Online]. Available: https://www.iso.org/obp/ui/#iso:std:iso:3386:-1:ed-2:v1:en

[9] “Qb SoftHand Research - anthropomorphic robotic hand - qbrobotics,” 1986, accessed: 2021-05-01. [Online]. Available: https://qbrobotics.com/products/qb-softhand-research/

[10] “UR10e Collaborative industrial robotic arm,” 2021, accessed: 2021-05-03. [Online]. Available: https://www.universal-robots.com/products/ur10-robot/

[11] “Qb robotics SoftHand,” Liberec, 2020, accessed: 2020-05-06. [Online]. Available: https://www.exactec.com/images/qb-hand-attack.jpg

[12] “Robots/qbhand - ROS Wiki,” 2018, accessed: 2021-01-05. [Online]. Available: https://wiki.ros.org/Robots/qbhand

[13] “qb SoftHand Research User Guide, version 1.0.0,” 2019, accessed: 2021-01-05. [Online]. Available: https://qbrobotics.com/wp-content/uploads/2016/03/qb-SoftHand-Research-User-Guide-1.0.0.pdf

[14] P. Mareš M., Stoudek P., 2021. [Online]. Available: https://gitlab.fel.cvut.cz/body-schema/ipalm/ipalm-grasping

[15] “Barrett Hand BH8-282,” 2021, accessed: 2021-05-01. [Online]. Available: https://advanced.barrett.com/barretthand

[16] “KUKA LBR iiwa,” 2021, accessed: 2021-05-01. [Online]. Available: https://www.kuka.com/en-de/products/robot-systems/industrial-robots/lbr-iiwa

[17] R. Navarro and J. Ariño, “barrett_controller – ROS Wiki,” 2014 (accessed April 4, 2020). [Online]. Available: http://wiki.ros.org/bhand_controller

[18] ——, “barrett_hand – ROS Wiki,” 2016 (accessed April 4, 2020). [Online]. Available: http://wiki.ros.org/barrett_hand

[19] ——, “rqt_bhand – ROS Wiki,” 2014 (accessed April 4, 2020). [Online]. Available: http://wiki.ros.org/rqt_bhand

[20] “RG6 gripper,” 2021, accessed: 2021-05-08. [Online]. Available: https://onrobot.com/en/products/rg6-gripper

[21] “RG6 gripper,” Trenčín, 2020, accessed: 2020-05-06. [Online]. Available: https://marvin-robotics.com/wp-content/uploads/rg6_gripper.jpg

[22] “OnRobot RG6 measurements,” accessed: 2020-04-25. [Online]. Available: https://drive.google.com/drive/folders/1qC1lW1cns17JHIVVWe8dnr3-RtXL8aZs

[23] "2F-85 - Robotiq," accessed: 2021-05-03. [Online]. Available: https://robotiq.com/products/2f85-140-adaptive-robot-gripper

[24] "Kinova Gen3 | Kinova," 2021, accessed: 2021-05-06. [Online]. Available: https://www.kinovarobotics.com/en/products/gen3-robot

[25] "Robotiq 2F-85," Doetinchem, 2020, accessed: 2020-05-06. [Online]. Available: https://wiredworkers.io/wp-content/uploads/2019/08/Robotiq-2-Finger-85-real.png

[26] "Robotiq 2F-85 measurements," accessed: 2020-04-28. [Online]. Available: https://drive.google.com/drive/folders/1oSHTesHJoGem_YGrsJkGIZFVPX0N8R70

[27] D. Subramanian, "A Simple Introduction to K-Nearest Neighbors Algorithm," 2018, accessed: 2020-05-04. [Online]. Available: https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e

[28] "Sklearn–Nearest Neighbors," accessed: 2020-03-28. [Online]. Available: https://scikit-learn.org/stable/modules/neighbors.html

[29] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020. [Online]. Available: https://doi.org/10.1038/s41586-020-2649-2

[30] "Sklearn–Standart Scaler," accessed: 2020-03-28. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[31] R. Gandhi, "Support Vector Machine — Introduction to Machine Learning Algorithms," 2018, accessed: 2020-05-03. [Online]. Available: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

[32] "Support vector machines (SVM)," accessed: 2020-05-18. [Online]. Available: https://is.muni.cz/el/1433/podzim2006/PA034/09_SVM.pdf

[33] "Sklearn–SVC," accessed: 2020-04-04. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[34] "Sklearn–GridSearchCV," accessed: 2020-04-04. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: https://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735

[36] C. Olah, "Understanding lstm networks," 2015 (accessed May 16, 2020). [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[37] M. Pliska, "Google drive," 2021. [Online]. Available: https://drive.google.com/drive/folders/1rK2hWKZBGhR3xaqBFZJdN7HHjYDie4Vf

[38] M. Scholz, "PCA - Principal Component Analysis," 2006, accessed: 2020-05-15. [Online]. Available: http://www.nlpca.org/pca_principal_component_analysis.html

[39] "Sklearn–PCA decomposition," accessed: 2020-04-16. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

[40] "qb SoftHand measurements," 2021, my measurements. [Online]. Available: https://drive.google.com/drive/folders/1OHqimGlFE6f6ivJRWyVo-RBKuiLMAMPM

[41] "Barrett Hand measurements," 2021, my measurements. [Online]. Available: https://drive.google.com/drive/folders/18DVUisKmyPgw3hqNXu47cgfnwgmh3Ega

[42] "Hand/282 – Barrett Technology Support," 2019 (accessed April 4, 2020). [Online]. Available: http://support.barrett.com/wiki/Hand/282

[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," 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. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[44] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017.

Appendix A

Classifiers' hyperparameters

A.0.1 qb SoftHand

	K
a1s1-squeez	1
a1s2-squeez	1
a2s1-squeez	1
a2s2-squeez	1
a1s1-squeez_and_release	1
a1s2-squeez_and_release	1
a2s1-squeez_and_release	1
a2s2-squeez_and_release	1

Table A.1: qb SoftHand - kNN

	K
a1s1-squeez	5
a1s2-squeez	5
a2s1-squeez	5
a2s2-squeez	5
a1s1-squeez_and_release	5
a1s2-squeez_and_release	5
a2s1-squeez_and_release	5
a2s2-squeez_and_release	5

Table A.2: qb SoftHand - kNN with features

Dataset	Kernel	C	γ
a1s1-squeez	linear	215.5	-
a1s2-squeez	linear	10	-
a2s1-squeez	linear	10	-
a2s2-squeez	linear	46.5	-
a1s1-squeez_and_release	rbf	1000	0.001
a1s2-squeez_and_release	linear	1000	-
a2s1-squeez_and_release	rbf	1000	0.01
a2s2-squeez_and_release	rbf	58.3	0.1
squeez	rbf	1000	0.1
squeez_and_release	rbf	1000	0.1

Table A.3: qb SoftHand - SVM

Dataset	Layers	Hidden dim	bs	Learning rate
a1s1-squeez	4	256	575	0.001
a1s2-squeez	4	256	604	0.0001854
a2s1-squeez	4	256	578	0.0008937
a2s2-squeez	4	256	539	0.001
a1s1-squeez_and_release	4	256	537	0.0004071
a1s2-squeez_and_release	4	256	586	0.0007139
a2s1-squeez_and_release	4	256	577	0.0001854
a2s2-squeez_and_release	4	256	502	0.0007139
squeez	4	256	1894	0.0002597
squeez_and_release	4	256	1849	0.0006380
squeez - pos ab	4	256	1894	0.0007988
squeez_and_release - pos ab	4	256	1849	0.0008937
squeez - curr ab	2	32	1894	0.0002906
squeez_and_release- curr ab	4	128	1849	1.7534865e-05

Table A.4: qb SoftHand - LSTM

A.0.2 Barrett Hand

Dataset	K
a1v0.6	1
a1v1.2	1
a3v0.6	1
a3v1.2	1
all	1

Table A.5: Barrett Hand - kNN

Dataset	K
a1v0.6	1
a1v1.2	1
a3v0.6	1
a3v1.2	1
all	1

Table A.6: Barrett Hand - kNN with features

Dataset	Kernel	C	γ
a1v0.6	linear	10	-
a1v1.2	linear	10	-
a3v0.6	linear	10	-
a3v1.2	linear	10	-
all	linear	10	-

Table A.7: Barrett Hand - SVM

Dataset	Layers	Hidden dimensions	Batch size	Learning rate
a1v0.6	2	256	128	0.00005
a1v1.2	2	256	128	0.00005
a3v0.6	2	256	128	0.00005
a3v1.2	2	256	128	0.00005
all	2	256	128	0.00005
all - tactile ablation	2	256	128	0.00005
all - effort ablation	2	256	128	0.00005

Table A.8: Barrett Hand - LSTM

A.0.3 OnRobot RG6

Dataset	K
objects	1

Table A.9: OnRobot RG6 - kNN

Dataset	K
objects	1

Table A.10: OnRobot RG6 - kNN with features

Dataset	Kernel	C	γ
objects	rbf	215.5	0.001

Table A.11: OnRobot RG6 - SVM

Dataset	Layers	Hidden dimensions	Batch size	Learning rate
objects	4	256	198	0.0002597
objects - width ab	2	32	198	0.0002597
objects - force ab	4	256	198	0.0005702

Table A.12: OnRobot RG6 - LSTM

A.0.4 Robotiq 2F-85

Dataset	K
objects	1
foams	3
foams-smaller-0068	4
foams-smaller-1445	4
foams-smaller-5085	4
foams-smaller-10000	4

Table A.13: Robotiq 2F-85 - kNN

Dataset	K
objects	1
foams	1
foams-smaller-0068	4
foams-smaller-1445	4
foams-smaller-5085	4
foams-smaller-10000	4

Table A.14: Robotiq 2F-85 - kNN with features

Dataset	Kernel	C	γ
objects	linear	10	-
foams	rbf	10	0.01
foams-smaller-0068	linear	10	-
foams-smaller-1445	linear	100	-
foams-smaller-5085	sigmoid	1000	-
foams-smaller-10000	sigmoid	46.5	-

Table A.15: Robotiq 2F-85 - SVM

Dataset	Layers	Hidden dim	bs	Learning rate
objects	4	256	244	0.0001657
foams	4	128	502	0.0002906
foams-smaller-0068	4	256	145	9.4538728e-05
foams-smaller-1445	4	128	158	3.4402132e-05
foams-smaller-5085	4	128	143	0.0005702
foams-smaller-10000	2	128	167	0.0003638
objects - pos ab	4	256	244	7.5517704e-05
foams - pos ab	4	128	502	6.7494485e-05
foams-smaller-0068 - pos ab	4	128	145	6.7494485e-05
objects - curr ab	4	256	244	8.4494661e-05
foams - curr ab	4	256	502	0.0001183
foams-smaller-0068 - curr ab	4	256	145	9.4538728e-05

Table A.16: Robotiq 2F-85 - LSTM

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Pliska Michal**	Personal ID number:	**483708**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Cybernetics and Robotics**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Deformable Object Classification Through Robot Grasping

Bachelor’s thesis title in Czech:

Klasifikace měkkých předmětů skrze mačkání robotickými uchopovači

Guidelines:

1. Familiarization with the gripper set: Barrett Hand (BH8-282, 3 fingers, 4 motors, 96 tactile sensors, 3 fingertip torque sensors), QB SoftHand (5 fingers, 1 motor, motor current sensor), 2-finger parallel grippers: Robotiq 2F-85 and OnRobot RG6 gripper (1 motor and current / force sensor).
2. Collect datasets by squeezing two deformable objects sets (9 everyday objects; 20 polyurethane foams) and recording sensory feedback. The objects sets have been already prepared and some data collected (Mareš 2020, Stoudek 2020). Grasping configuration and squeezing speed should be varied.
3. Unsupervised analysis of time series data. Clustering of time series in order to study whether the principal clusters formed will be dominated by the differences in objects or by factors related to gripper or grasping action parameters. Multivariate Singular Spectrum Analysis followed by k-means clustering is a possibility. Assess whether material properties like stiffness can be extracted (Liarokapis et al. 2015).
4. Object classification from collected time series using LSTM neural network (follow up on Mareš 2020).
5. Assess the effect of gripper type, action parameters (gripper configuration and squeezing speed), sensory channels availability (including ablation experiments), and LSTM network size on recognition results.

Bibliography / sources:

[1] Bednarek, J., Bednarek, M., Kicki, P., & Walas, K. (2019). Robotic Touch: Classification of Materials for Manipulation and Walking. In 2019 2nd IEEE International Conference on Soft Robotics (RoboSoft) (pp. 527-533). IEEE.
[2] Hoffmann, M., Stepanova, K. & Reinstein, M. (2014), 'The effect of motor action and different sensory modalities on terrain classification in a quadruped robot running with multiple gaits', Robotics and Autonomous Systems 62, 1790-1798.
[3] Mareš, M. (2020), 'Exploratory Action Selection to Learn Object Properties from Haptic Exploration Using a Robot Hand', Bachelor's thesis, Czech Technical University, Faculty of Electrical Engineering.
[4] Liarokapis, M. V., Calli, B., Spiers, A. J., & Dollar, A. M. (2015, September). Unplanned, model-free, single grasp object classification with underactuated hands and force sensors. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5073-5080). IEEE.
[5] Sanchez, J., Corrales, J. A., Bouzgarrou, B. C., & Mezouar, Y. (2018). Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey. The International Journal of Robotics Research, 37(7), 688-716.
[6] Stoudek, P. (2020), 'Extracting material properties of objects from haptic exploration using multiple robotic grippers', Master's thesis, Czech Technical University, Faculty of Electrical Engineering.

Name and workplace of bachelor’s thesis supervisor:

Mgr. Matěj Hoffmann, Ph.D., Vision for Robotics and Autonomous Systems, FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Ing. Zdeněk Straka, Vision for Robotics and Autonomous Systems, FEE

Date of bachelor’s thesis assignment: **06.01.2021** Deadline for bachelor thesis submission: **21.05.2021**

Assignment valid until: **30.09.2022**

Mgr. Matěj Hoffmann, Ph.D.	prof. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature	Head of department’s signature	Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

.	
Date of assignment receipt	Student’s signature