Accepted in IEEE International Conference on Robotics and Automation (ICRA), 2017

Grasp Stability Assessment through Unsupervised Feature Learning of Tactile Images

Deen Cockburn, Jean-Philippe Roberge, Thuy-Hong-Loan Le, Alexis Maslyczyk and Vincent Duchaine

***Abstract*— Grasping tasks have always been challenging for robots, despite recent innovations in vision-based algorithms and object-specific training. If robots are to match human abilities and learn to pick up never-before-seen objects, they must combine vision with tactile sensing. This paper present a novel way to improve robotic grasping: by using tactile sensors and an unsupervised feature-learning approach, a robot can find the common denominators behind successful and failed grasps, and use this knowledge to predict whether a grasp attempt will succeed or fail. This method is promising as it uses only high-level features from two tactile sensors to evaluate grasp quality, and works for the training set as well as for new objects. In total, using a total of 54 different objects, our system recognized grasp failure 83.70% of time.**

I. INTRODUCTION

Robots have been practicing tasks like object grasping and manipulation for the past 50 years, but they are still far from matching human capabilities in this domain. The ability to grasp a never-before-seen object—something humans develop in their first year of life [1]—remains a major challenge for robots. In the past, production lines required robots to pick up the same object hundreds of times. Today, however, the lean supply chain movement (characterized by small volume and high mix production) means that robots must learn to handle a variety of objects.

One way of improving grasp performance is to develop better robotic vision. Researchers at Brown [2] have begun collecting a database of 3D image scans of various objects to make it easier for robots to recognize them. More recently, a team at Google [3] combined visual data with machine learning in an attempt to have robots teach themselves how best to pick up various objects. These attempts are promising, but vision-based grasping systems are limited by the difficulties posed by certain conditions, such as when there is low light, a reflective or translucent surface, or a partially-obscured target object (e.g. in bin-picking tasks). More importantly, grasping is by nature about contact and forces, things that cannot easily be seen with vision. So although vision-based systems can determine a grasping strategy that is likely to succeed, vision alone is unlikely to match the performance of a system that also uses tactile information.

Recent years have seen numerous efforts to provide robots with tactile sensors [4], [5], [6], [7]. The sense of touch is the key to humans' astounding grasp abilities, and extending this sense to robots could greatly improve robots' skills. However, developing the right tactile sensor is only part of the solution: it must be accompanied by an intelligent algorithm, to make sure the sensors' complex output signals are translated into better grasping abilities.

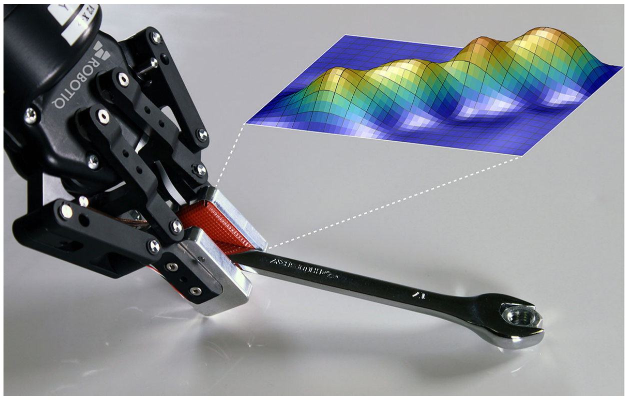


Fig. 1. Tactile sensors on Robotiq gripper

Tactile intelligence can be broken down into numerous components. For instance, researchers have used tactile sensors to recognize object textures [8], and to detect object slippage [9], [10], [11]. Another component would be the ability to evaluate grasp stability, and use this knowledge to predict grasp failure. The goal is for robots to one day have the same capacity as humans for knowing whether they have a solid grasp on an object, and adjusting their grasps automatically if they do not feel stable.

Several works have brought the robotics community closer to this goal. Some researchers have studied this problem by starting from a mathematical definition of grasp stability [12], [13] that uses the direction of the force vector at the contact point to predict the best grasp strategy. Other researchers have tried to use these grasp quality metrics in conjunction with tactile sensing to assess grasp quality in real time [14]. Whereas these two studies based their grasp quality assessments on analytic stability criteria, other researchers used machine learning, based on hand-crafted features, to recognize a stable grasp [15]. More recently, another team [16] has built an object-based intelligence that can predict whether a grasp attempt will succeed or fail. By gathering object-specific knowledge, in terms of knowing what constitutes a successful vs. unsuccessful grasp for each object, they have achieved impressively accurate predictions (up to 90%) on their object set.

Although an object-specific prediction algorithm could probably help robots reach near-perfect grasping abilities for

The authors are with the Command and Robotics Laboratory (CoRo), Faculty of Automated Production Engineering, École de technologie supérieure, Montréal, Québec, Canada

Corresponding author: vincent.duchaine@etsmtl.ca objects they have trained with, we believe another method is needed if robots are to have the same success with never-before-seen objects. In this paper, we explore whether successful (and failed) grasps share a common denominator that could allow robots to predict, using an SVM classifier, the grasp outcome for any object. We did not use handcrafted features to find these common denominators. Instead, we used an unsupervised feature-learning approach that gave a high-level representation of what the system found to be the relevant features.

We begin by presenting the tactile sensor we used in this work (see Fig. 1). Section 3 describes our approach, including the auto-encoder algorithm for feature-learning and the system's optimization process. In Section 4 we detail the experimental procedure, with results given at the end. Finally, Section 5 concludes the paper with an analysis of the features that were learned by the robot.

II. Tactile Sensor

We developed our grasp stability prediction system using only the pressure images from tactile sensors as our input. The sensor we used was developed in our laboratory, the Command and Robotics Laboratory (CoRo) at École de technologie supérieure (ÉTS). The sensor can measure both static (pressure images) and dynamic variations in pressure over time, although only the first type of information is used in this work. The device relies on capacitive sensing to acquire both types of data, with pressure data acquired at a rate of 25 Hz with a resolution of 4×7 taxels. The sensor has a wide measurement range and relatively high sensitivity because of its micro-structured polyurethane dielectric. The dielectric was built using a direct laser-etching technique, which makes it quick and inexpensive to fabricate, unlike the moulded dielectric in our lab's earlier version of the sensor [17]. The sensor can withstand up to 400 kPa. Since a typical human grasp is between 10-100 kPa [18], this is more than enough for our "everyday" grasping needs.

III. Proposed Approach

The goal of this work is to improve robotic grasping by enabling a robot to distinguish between stable and unstable grasps for a variety of objects. To achieve this, we are proposing an approach that lets our system find the features of tactile images that are most relevant for the task of predicting whether a grasp attempt will succeed or fail. Our grasp analysis method is based on pressure images captured by a tactile sensor. The original aspect of our work comes from the fact that we used an unsupervised feature learning algorithm to achieve our goal, rather than hand-crafting the features.

In the past, several researchers have improved their robots' abilities to grasp a variety of objects by hand-crafting features according to pressure image moments [19], [15] or inspiration from human tactile sensing [20]. However, the success of hand-crafted feature techniques is entirely reliant upon the researchers' abilities to determine the most relevant features.

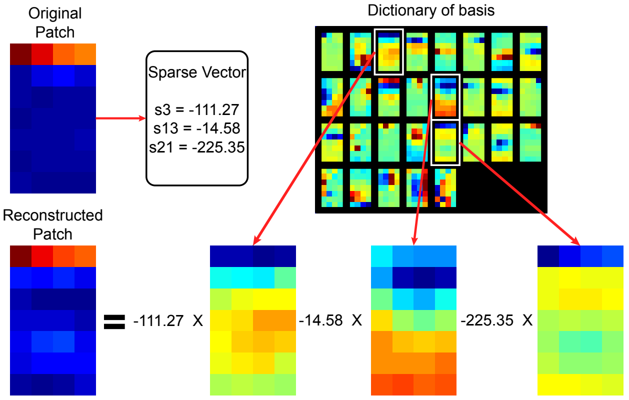


Fig. 2. Reconstruction of a patch

By contrast, in our approach the auto-encoder (specifically a sparse coding algorithm) itself determines the most relevant high-level features of the unlabeled pressure image data. These high-level features are then used to classify the pressure image data with an SVM. The SVM classifier chooses the most relevant features (from among the high-level features) for distinguishing between the two groups, a task to which this type of classifier is well-suited [21]. By encoding the data and finding most relevant high-level features, we are hypothesizing that this will lead to knowledge of the combination of high-level features that most strongly correlates with the group of successful grasps (and likewise for the group of failed grasps). Thus the algorithm and SVM are working together to find the common denominators behind all successful (and all failed) grasps.

A. Data Auto-Encoding

In this section, we describe the techniques we used to encode the raw data for automatic feature extraction, and then explain the approach we used to optimize the prediction algorithm. Much like previous work [19], [15], we consider our static tactile pressure data to be an image. Here, we use *tactile image* to refer to the two pressure images from the sensors that were recorded at the moment of the grasp and placed side-by-side to make one composite pressure image (an example of a filtered sensor image can be seen in Fig. 12).

To give a theoretical overview of sparse coding, it works by creating a dictionary of *basis vectors*. Each basis vector is a high-level feature of the input data, and they are used to reconstruct the original image. In other words, the dictionary is used to represent our original tactile image patches as a linear combination of the dictionary's basis vectors. Our sparse coding approach uses image patches that follow the format of our tactile sensors. Since we combined our two pressure images, two patches are needed to reconstruct the pressure image. Our resulting dictionary is composed of basis vectors of dimension 28 (4×7).

Let $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)} \in \mathbb{R}^k$ be the m patches of a certain tactile image $\boldsymbol{X} \in \mathbb{R}^{k \times m}$, such that each patch has k taxel intensity values. The idea is to find a sparse vector $\boldsymbol{\alpha}^{(i)} \in \mathbb{R}^n$ for each $\boldsymbol{x}^{(i)}$ by using some *a priori* learned basis from a dictionary

$\boldsymbol{D} := [\boldsymbol{d}_1, ..., \boldsymbol{d}_n] \in \mathbb{R}^{k \times n}$, such that:

$$\boldsymbol{x}^{(i)} \approx \sum_{j=1}^{n} \boldsymbol{d}_j \alpha_j^{(i)} \qquad i = 1, ..., m. \tag{1}$$

To obtain the sparse vectors $\boldsymbol{\alpha}^{(1)}, ..., \boldsymbol{\alpha}^{(m)}$ that capture high-level features of $\boldsymbol{X}$ in the dictionary, the dictionary of basis $\boldsymbol{D}$ must first be learned. This is done by minimizing the following objective function:

$$\min_{\boldsymbol{D},\boldsymbol{\alpha}} \sum_{i=1}^{m} \left(\left\| \boldsymbol{x}^{(i)} - \sum_{j=1}^{n} \boldsymbol{d}_j \alpha_j^{(i)} \right\|^2 + \beta \sum_{j=1}^{n} ((\alpha_j^{(i)})^2 + \varepsilon)^{1/2} \right). \tag{2}$$

The first term of eq. 2 inside the summation is the squared representation error, thus penalizing the objective function for poor representation of the input vectors. Regarding the second term, β is an arbitrarily-set scalar that will define the importance of *sparsity*. Sparsity is the ratio between the quantity of active basis vectors to the number of basis vectors in the dictionary. A high sparsity corresponds to a low amount of active basis vectors. This second term penalizes the objective function when non-sparsity is high, and thus is responsible for making each $\boldsymbol{\alpha}^{(i)}$ sparse.

The double minimization problem stated in eq. 2, is a complex one that is known to be computationally expensive. However, [22] has shown that it can be split into two convex optimization problems, which can then be solved iteratively. Moreover, dictionary learning can be performed offline so that it does not affect live operations.

When the dictionary $\boldsymbol{D}$ is complete it can be used to represent our data. In order to determine which basis must be used and with which intensity, the following equation must be minimized:

$$\min_{\boldsymbol{\alpha}} \left\| \boldsymbol{x} - \sum_{j=1}^{n} \boldsymbol{d}_j \alpha_j \right\|^2 + \gamma \sum_{j=1}^{n} |\alpha_j|. \tag{3}$$

To solve these mathematical problems, we used the MATLAB code made available by [22]. Fig. 2 illustrates how the dictionary is used to reconstruct a patch of our sensor image. The sparse vector α is obtained using eq. 3.

The result of sparse coding are a set of sparse vectors composed of the coefficients for each basis vector needed to reconstruct the input vector. This can also be interpreted as a decomposition of our input vector into high level features.

B. Optimization process

Sparse coding is a double optimization problem that has the objective of reconstructing an image as best it can under the constraint of using a limited amount of elements in a dictionary. Since the "optimal" dictionary is the one that yields the best classification results under the constraints of its sparse coding parameters, there are an infinite amount of optimal dictionaries because there are infinite variations of sparse coding parameters. Each of these "optimal" dictionaries will lead to different classification results. Our goal is to find the sparse coding parameters that will result in, as closely as possible, the best classification of our grasp data into successful and failed categories.

There are many parameters in the sparse coding algorithm, as shown in the previous subsection. We chose to focus on only three of these parameters: the size of the dictionary (num_bases), the sparsity penalty factor of the dictionary construction (β), and the sparsity penalty factor used during the reconstruction phase (γ). For computational reasons, the dictionary learning and reconstruction phases use different sparsity penalties. The *epsilonL1* penalty function is used during dictionary learning (eq. 2), whereas the *L1* penalty function is used during reconstruction (eq. 3). Allowing differences between the two penalty factors (β and γ) enables the optimization process to have different sparsity levels during the two phases.

Since there is no straightforward closed-form solution that determines the optimal dictionary parameters (num_bases and β) and the optimal sparsity during the reconstruction phase ((γ), we used a brute force approach: the grid search method. With every iteration, we modified one parameter and computed the results. The optimization algorithm steps are the following: first, generate the optimal dictionary of basis using the efficient sparse coding algorithm; second, encode all the raw data using the reconstruction algorithm explained in section III-A; finally, use a ten-fold cross validation to train linear SVMs to compute a weighted success rate (see Fig. 3). For every iteration, we saved all the data generated by the process.

The following is the weighted success rate used in the third step of the optimization process:

$$WeightedSuccessRate = \left(\frac{CCS}{TS} + \frac{CCF}{TF} \right) * \frac{100\%}{2}. \tag{4}$$

This weighted success rate compensates for the fact that of the labeled data, the grasp successes outnumber grasp failures. It works by computing the ratios between correctly classified data (CCS and CCF) and the total data (TS and TF) for both labels. Often it is better to have a false negative (to incorrectly classify a success as a fail) than to have a false positive (to incorrectly classify a fail as a success). By applying this simple equation to our data, we give equal importance to correctly classified successes and correctly classified fails.

The results of this automated testing algorithm were then used to determine the optimal parameters for our system by extracting the best weighted success rates. The span and step for every variable were determined by manual testing of

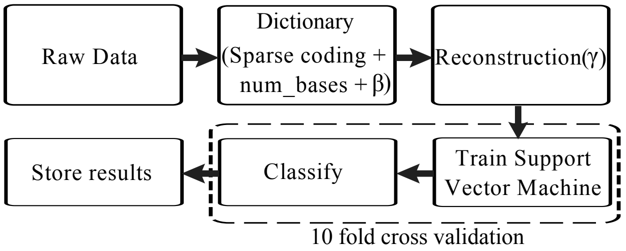

Raw Data
Dictionary (Sparse coding + num_bases + β)
Reconstruction(γ)
Train Support Vector Machine
Classify
Store results
10 fold cross validation

Fig. 3. Optimization process

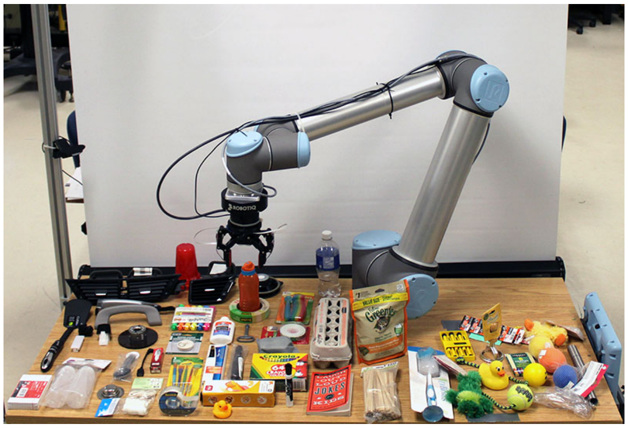


Fig. 4. The experimental setup and 54 objects used

different combinations of parameters prior to launching the automated testing algorithm.

IV. EXPERIMENTATION

Here we describe the experimental setup we used to acquire the data for both optimizing the system's parameters and for training the classifiers. Before, we needed a variety of unlabeled data for sparse coding. Now, to validate our work, we need a representative set of labeled data to ensure proper training and testing of our classifier.

A. Experimental setup

In order to obtain the most representative data, we used our automated picking station (shown in Fig. 4). First, we installed two tactile sensors (described in section II) on a 2-finger gripper from Robotiq, Inc, which is an underactuated adaptive parallel gripper. The underactuation of the fingers allows us to use a variety of objects that could not be picked up by a rigid parallel gripper. However, the underactuation can also lead to unpredictable events, as will be discussed later. This gripper, along with a force/torque sensor (also by Robotiq Inc.), was integrated with a UR10 manipulator from Universal Robot. The UR10 has six independent rotational joints and a maximum payload of 10 kg.

We included a vision system so the robot could locate the object to pick up. A Microsoft Kinect© was used to collect the vision data. Our vision algorithm located the object and returned the position of the geometrical center of the object. Using this data, we computed a trajectory for a simple top pick aiming at the center of the object.

We defined a $successful$ grasp as such: the robot must pick the object up out of the bin, put the object through a dynamic motion test and then drop it back in the bin. A $failed$ grasp occurs when the object is not placed back in the bin. The dynamic tests consisted of various patterns of rapid acceleration and deceleration of the robot in a manner that would likely cause it to drop the object (vertically up and down, vertical swinging, forward and back). During this test, the robot's end effector had a speed of $0.6m/s$ and accelerations of $1.2m/s^2$.

The entire system was controlled via an ROS framework running Ubuntu. The system included a datalogging module to record the tactile sensors' pressure images. It is important to specify that these pressure images were taken once, when the gripper was in full contact with the object. This setup was used to collect data from a total of 54 everyday objects. Each object was picked 10 times for a total of 540 picks.

B. Experimental results

To validate our approach, we ran the optimization algorithm described in section III-B with the collected data. To compare each dictionary size (variation on the number of basis), we extracted the best result for each size by varying the other two parameters. The resulting weighted success rates for the best parameter combination at each basis size are plotted in Fig. 5. One can observe that the weighted success rates increase sharply until around an 11 basis dictionary. From this point, only small variations are computed. The overall best result is obtained with a dictionary of 29 basis.

This dictionary can be visualized in its raw format in Fig. 6(a). We also added a filtered version of the dictionary in Fig. 6(b) to help us understand the features represented by each basis. Observing the figures, we can surmise some attributes of the high-level features created by the sparse coding algorithm. For example, we can see in the third (3) basis of the dictionary a feature that may be described as an edge contact made at the tip of the sensor. Fig. 2 depicts the reconstruction phase of the sparse coding process. We note that our optimal dictionary is composed of 29 basis, whereas our original tactile image is composed of 28 taxels, so our optimal dictionary corresponds to the first iteration of an overcomplete dictionary.

When we used this dictionary to create sparse vectors for all of our raw data, we obtained an average sparsity of ***86.31%***. We used a 10-fold cross validation on our 540 sparse vectors to verify the classification efficiency of the simple linear SVMs. Considering that we do not have an equal amount of successful and failed picks in our data, we used a weighted success rate to evaluate the efficiency of our classifier. We obtained a ***78.89%*** weighted success rate with the following parameters: 29 basis in the dictionary, a penalty on the dictionary optimization process of $\beta = 950$ and a sparsity penalty on the reconstruction process of $\gamma = 1000$. In comparison, [16] acheived a better success rate of 89%

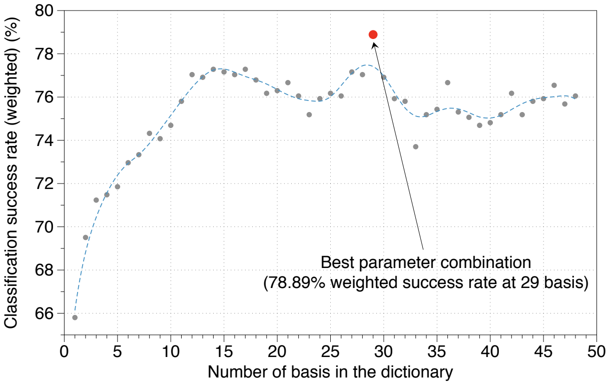
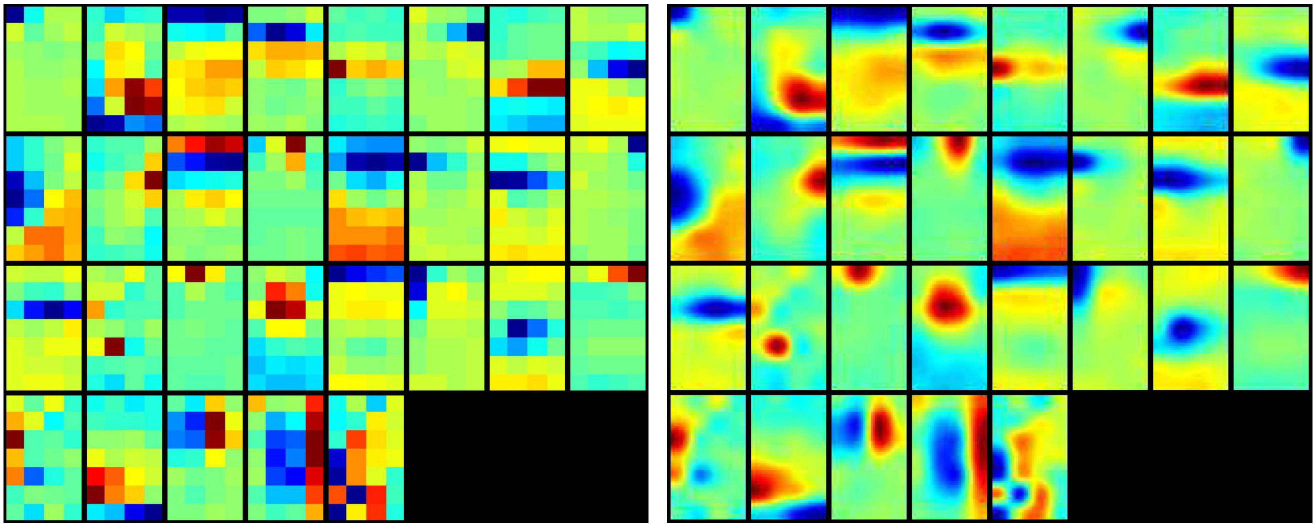


Fig. 5. Success rate vs. number of basis



(a) Our optimal dictionary

(b) Filtered dictionary

Fig. 6. Dictionary, result of optimal sparse coding

but, by using a higher number of inputs. In fact, our system uses only two tactile sensors as an input while their system relies on tactile information, hand configuration (joint angles) and 3d shape data derived from a vision system input. Therefore, we are confident that we could probably get better results by adding additional inputs to our system.

To further understand the performance of our classifiers, Fig. 7 shows the confusion map of our results. The success rate for classifying failed grasps is ***83.70%***. We would like to point out that we consider the success rate for failed grasps to be one of the most important indicators of the system's performance. We included the correctly classified successes in our weighted success rate because it was necessary for having the robot attempt the grasps. Otherwise, since we wanted to prioritize not dropping the object, but we did not place the same importance on not aborting a potentially successful grasp, the robot would logically decide to attempt fewer grasps, which would make for a frustratingly inefficient robot.

V. Discussion

A. *Sparse coding analysis*

Since the algorithm needs input data that fully represents our population, one may ask how much data is needed for a statistically sound representation of the latter. In our problem, the population is very hard to represent considering we want a system that can grasp any object that can fit in our gripper. Seemingly by coincidence, our labels were separated perfectly into 75% successful and 25% failed picks: we had 405 examples of successful picks and 135 examples of failed picks. Since our priority is to capture failed grasps, we most likely could improve our results by adding more data from failed picks.

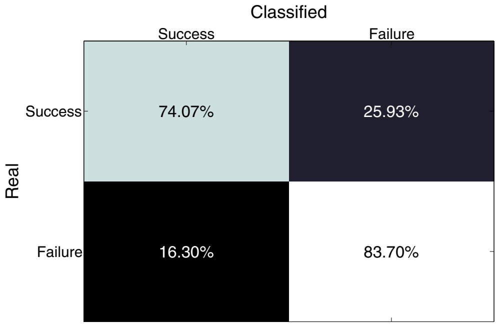


Fig. 7. Confusion map

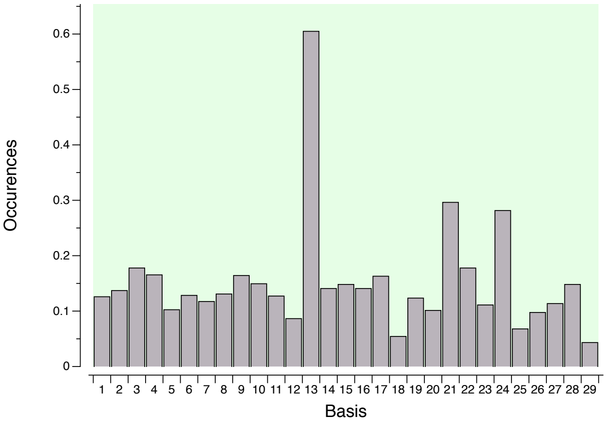


Fig. 8. Successful picks basis usage

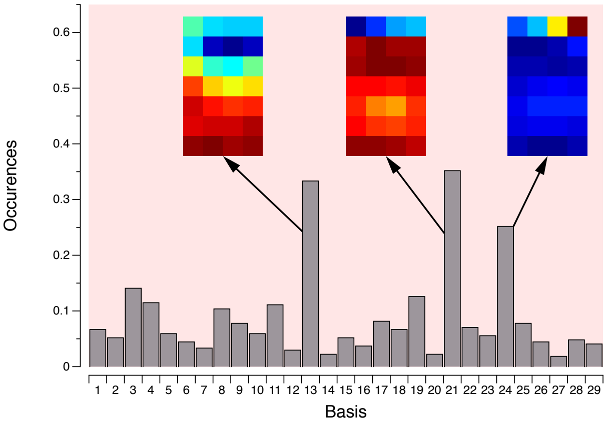


Fig. 9. Failed picks basis usage

Also, as shown in eq. 3 we included a *sparsity penalty* which translates into using as few basis from the dictionary as possible (given the other parameters) to represent each patch of our image. We computed the average quantity of basis used to represent successes and fails. On average, 4.4259 basis are used to represent a successful pick patch and 2.5926 basis are used to represent a failed pick patch. By observing the tactile images, we saw that most successful pick images had larger active areas (compared to failed pick images), indicating that more of the sensor was in contact with the object during successful picks. We can infer from this that successful pick images require more basis for their reconstruction than failed pick images.

This led us to wonder whether certain basis of the dictionary are used mainly to represent one class of picks. In Fig. 8 and Fig. 9 we computed the weighted occurrences for every basis of the dictionary by label. Each bar is a normalized occurrence of the basis by label. For example in Fig. 8, basis 13 is at approximately 60% occurrence, meaning that this basis was involved in the representations of 60% of successful picks. Unexpectedly, the same three basis are used more then 25% of the time for both successful and failed picks. These basis are extracted in Fig. 9. If we analyze the basis in this figure, we notice that they are mostly fingertip edge contact features, but we still do not know how these features help the classifier.

We decided to push this analysis further. We wished to know whether some of the basis are used mainly for one label or the other. To find out, we computed the differential of the two graphs in Fig. 8 and Fig. 9, with the results shown in Fig. 10. If the bars are in the left section of the graph, the basis are more present in failed picks, and vice versa. In this case, we see a new set of basis that are most often used for failed picks. Only basis #21 is in both sets (used most often for failed picks in both the original graph and the differential graph). This basis is used to construct a pressure image of a pick on the very edge of the fingertip, whereas the other basis of this analysis seems to represent very weak contact points. If one were to try and hand-craft features, we speculate that #21 would probably be one of the basis to consider using.

Here we investigate the role of coefficients. In Fig. 11 we can see the differential of the absolute coefficient averages per basis. This time, if the bar is towards the left of the graph, we can say that the basis has a larger coefficient value when used to represent a failed grasp. We notice that failed grasps are more likely to have a strong coefficient. If we couple this information with the fact that failed picks are usually represented using fewer basis, we could hypothesize that a strong activation of a few specific basis would allow us to classify the pick as a fail.

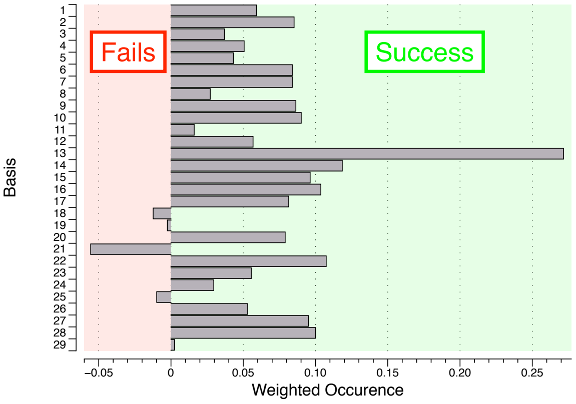


Fig. 10. Basis usage differential

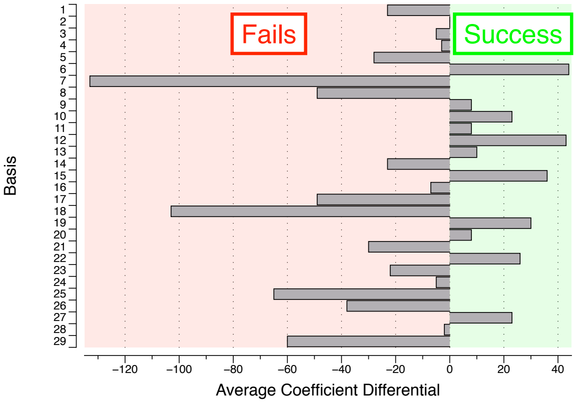
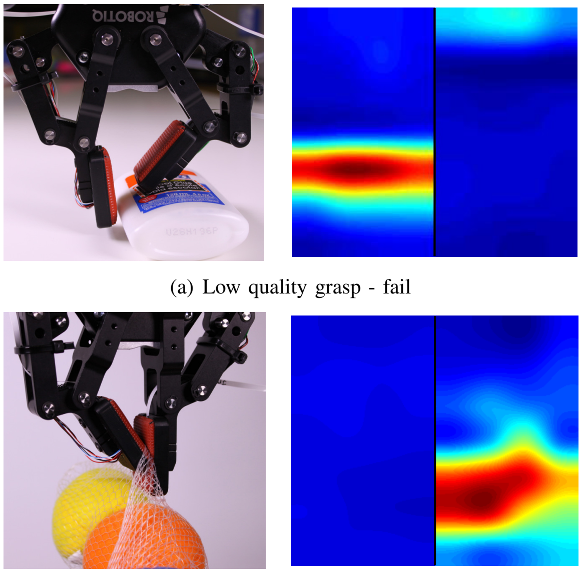


Fig. 11. Basis Average Coefficient Differential

B. *The classifier's performance analysis*

In addition to our main goal, we also hoped to find the common denominators behind successful and failed grasps for all objects. The fact that a simple linear classifier like an SVM can separate successful and failed grasps for 54 objects (with 78.89% weighted success rate) is encouraging because it indicates that other objects could possibly be classified similarly. However, the way the k-fold was performed (the data were shuffled randomly) means we cannot be sure that the test data sets were from objects not used for the data in the training sets. To address this concern, we performed the same experiment with 50 completely new objects. These objects are similar to those presented in Fig. 1, such as the ones presented in [23]. The results of this new experiment rendered a weighted success rate of 71.36%. Although we did not reach the same level of performance, the results are encouraging. They are especially promising when we consider that we used a new set of tactile sensors right after the first experiment, putting even more at test the generalization capacity of the high-level features contained in the dictionary.

Lastly, as mentioned earlier, we noticed that the underactuation of the gripper can create some confusion in the data (one example can be seen in Figs. 12(a) and 12(b)). In both cases, if we only concentrate on the pressure images, they can seem very similar. Our algorithm only uses the static pressure images, so in future work we might include more information to potentially get better classification results. We will attempt to integrate some gripper information into the algorithm, such as by using the integrated IMUs to compute the finger positions in 3-dimensional space. Therefore, we will study the different data fusion techniques to hopefully correctly classify this confusing data.



(a) Low quality grasp - fail

(b) Low quality grasp - success

Fig. 12. Low quality grasp examples

VI. CONCLUSIONS

To improve their manipulation skills and learn to pick up never-before-seen objects, robots must have greater tactile sensing capacities. But the solution is not simply to develop more advanced tactile sensors, because robots also need a way to translate the sensor's signals into information they can act upon. One way in which tactile sensors can be used to improve manipulation skills is by assessing grasp stability according to tactile images. Eventually, robots could use their prediction of whether the grasp will fail to adjust their grip in real time.

In this paper, we have explored whether successful (and failed) grasp attempts share a common denominator that could allow robots to predict the grasp outcome for any object. We proposed an approach to grasp assessment that used unsupervised feature learning to find the most relevant high-level features for distinguishing between tactile images of successful and failed picks, via sparse coding and a simple linear SVM. In two experiments with 54 and 50 objects, respectively, we demonstrated that our system works well with both the training set and never-before-seen objects. In total, with 54 different objects our system had a weighted success rate of 78.89%. With 50 new objects the weighted success rate was 71.36%.

REFERENCES

[1] T. Hinojosa, C.-F. Sheu, and G. F. Michel, "Infant hand-use preferences for grasping objects contributes to the development of a hand-use preference for manipulating objects," *Developmental Psychobiology*, vol. 43, no. 4, pp. 328–334, 2003.

[2] J. Oberlin and S. Tellex, "Autonomously acquiring instance-based object models from experience," *Int. S. Robotics Research (ISRR)*, 2015.

[3] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *CoRR*, vol. abs/1603.02199, 2016. [Online]. Available: http://arxiv.org/abs/1603.02199

[4] J. Engel, J. Chen, and C. Liu, "Development of polyimide flexible tactile sensor skin," *Journal of Micromechanics and Microengineering*, vol. 13, no. 3, p. 359, 2003.

[5] A. Schmitz, M. Maggiali, L. Natale, B. Bonino, and G. Metta, "A tactile sensor for the fingertips of the humanoid robot icub," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 2212–2217.

[6] N. Wettels, J. Fishel, Z. Su, C. Lin, and G. Loeb, "Multi-modal synergistic tactile sensing," in *Tactile sensing in humanoidsTactile sensors and beyond workshop, 9th IEEE-RAS international conference on humanoid robots*, 2009.

[7] A. Rana, J.-P. Roberge, and V. Duchaine, "An improved soft dielectric for a highly sensitive capacitive tactile sensor," *IEEE Sensors Journal*, vol. 20, 2016.

[8] F. De Boissieu, C. Godin, B. Guilhamat, D. David, C. Serviere, and D. Baudois, "Tactile texture recognition with a 3-axial force mems integrated artificial finger." in *Robotics: Science and Systems*. Seattle, WA, 2009, pp. 49–56.

[9] M. R. Tremblay and M. R. Cutkosky, "Estimating friction using incipient slip sensing during a manipulation task," in *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*. IEEE, 1993, pp. 429–434.

[10] B. Heyneman and M. R. Cutkosky, "Biologically inspired tactile classification of object-hand and object-world interactions," in *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, Dec 2012, pp. 167–173.

[11] J.-P. Roberge, S. Rispal, T. Wong, and V. Duchaine, "Unsupervised feature learning for classifying dynamic tactile events using sparse coding," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 2675–2681.

[12] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics Automation Magazine*, vol. 11, no. 4, pp. 110–122, Dec 2004.

[13] G. A. Kragten and J. L. Herder, "The ability of underactuated hands to grasp and hold objects," *Mechanism and Machine Theory*, vol. 45, no. 3, pp. 408–425, 2010.

[14] H. Dang, J. Weisz, and P. K. Allen, "Blind grasping: Stable robotic grasping using tactile feedback and hand kinematics," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5917–5922.

[15] Y. Bekiroglu, J. Laaksonen, J. A. Jorgensen, V. Kyrki, and D. Kragic, "Assessing grasp stability based on learning and haptic data," *IEEE Transactions on Robotics*, vol. 27, no. 3, pp. 616–629, June 2011.

[16] E. Hyttinen, D. Kragic, and R. Detry, "Learning the tactile signatures of prototypical object parts for robust part-based grasping of novel objects," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 4927–4932.

[17] V. Duchaine and A. Rana, "Dielectric geometry for capacitive-based tactile sensor," July 24 2014, wO Patent App. PCT/CA2014/050,040. [Online]. Available: http://www.google.se/patents/WO2014110683A1?cl=sv

[18] J. Dargahi and S. Najarian, "Human tactile perception as a standard for artificial tactile sensinga review," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 1, no. 1, pp. 23–35, 2004.

[19] Y. Bekiroglu, R. Detry, and D. Kragic, "Learning tactile characterizations of object- and pose-specific grasps," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2011, pp. 1554–1560.

[20] J. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *Robotics, IEEE Transactions on*, vol. 27, no. 6, pp. 1067–1079, Dec 2011.

[21] J. Mairal, F. R. Bach, and J. Ponce, "Sparse modeling for image and vision processing," *CoRR*, vol. abs/1411.3230, 2014. [Online]. Available: http://arxiv.org/abs/1411.3230

[22] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 801–808. [Online]. Available: http://papers.nips.cc/paper/2979-efficient-sparse-coding-algorithms.pdf

[23] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *Advanced Robotics (ICAR), 2015 International Conference on*. IEEE, 2015, pp. 510–517.