Machine Learning for Camera-less Object Classification with Soft Robot Grippers

HARVARD
JOHN A. PAULSON
SCHOOL OF ENGINEERING
AND APPLIED SCIENCES

Matheus C. Fernandes*[1], Andrew Gross[1], Nikolaos Vasios[1], Katia Bertoldi[1,2] (PI)
[1]John A. Paulson School of Engineering and Applied Sciences, Harvard University | [2]Kavli Institute, Harvard University

IMECE 2017-72882
NSF Graduate Poster Competition

Project Objectives and Goals

Overarching Goal:

Build and autonomously operate a Soft Robotic Gripper which can detect and classify objects solely based on data from arrays of embedded sensors using Artificial Intelligence.

Individual Objectives:

- We propose the development of an autonomous soft robotic system with distributed sensors that is able to perform object recognition via sensor data.
- We study different Machine Learning algorithms to perform the task of efficiently and effectively classifying objects and detecting arbitrary shapes from reasonable sized training sets.
- The system is to serve to serve not only as a proof of concept, but also as a principle for designing similar machinery when using additional types of embedded sensors.

Background

Soft Robotics:

- First, what is a 'Soft Robot'?
  - It is exactly what it sounds like, it's a robot that is constructed using highly compliant materials similar to those found in living organisms.
  - Soft robotics allows for increased flexibility and adaptability to accomplish various tasks.
  - It is especially advantageous over 'conventional robots' in applications where handling fragile or sensitive objects such as eggs, fruits, vegetables etc.
- Soft robotic actuators have become a topic of intense research and has gained the interest of many industrial groups.

Artificial Intelligence (Machine Learning):

- Convolutional Neural Networks are widely used for image recognition and are useful in finding and learning patterns of an objects shape and size. With large enough sensor resolution, a similar technique could be incorporated into our model.
- For sensor data, we can use the same basic AI principles to learn distinct features from the data as done with camera feeds.
- Exploiting widely used Machine Learning algorithms we can learn patterns from an array of sensor data and statistically minimize the effect of noise in our inference prediction.

Motivations:

- Inference on classification of objects via projection methods, such as camera data, is limited by the cameras inability to obtain physical measurements such as compliance of an object. Furthermore, to obtain dimensionally independent classification, triangulation algorithms are required and are limited by each cameras field of view.
- Data such as the precise state of an actuator is used in conventional robotic systems to infer information about the objects that they are interacting with, and is currently lost in soft robotic systems.

Applications:

- With the rise in manufacturing automation as well as the delivery and autonomous grocery store market, the demand for such classifying soft actuators is on a steep rise.

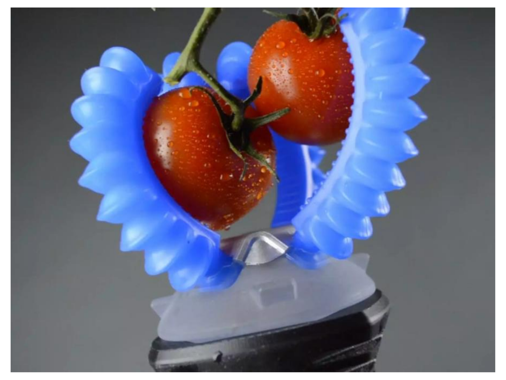

Fig 1: Pneumatic gripper

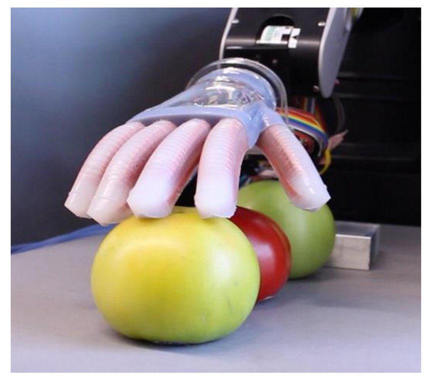

Fig 2: Hand-like pneumatic gripper

Fig 1 and Fig 2 courtesy of https://softroboticstoolkit.com/

Experimental Setup

Actuator System:

A pneumatic actuator is used as the main gripping mechanism for the system. A volume controlled pump feeds air into the actuator and allows the actuator to bend. The actuator itself is composed of a multi-chamber array that bends upon inflation, exerting force on any objects that may be in contact with it. (Fig. 3)

Data Acquisition System:

A Raspberry Pi acts as the centralized brain of the system and data from the sensors attached to the actuator feed into the Pi. There are a total of 9 sensors on the actuator.

Attached Sensors (illustrated in Fig. 4):

- 4 resistive curvature sensors
- 4 resistive force sensors
- 1 pressure sensor

Test Objects (seen in Fig.6):

- Stiff Cylindrical Can
- Stiff Square Cross Section Can
- Compliant Roll of paper
- Compliant Spherical Stress Ball

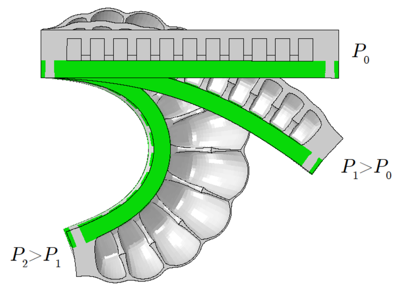

Fig 3: Inflating actuator rendering.

Fig 4: Deflated actuator with sensors. Sensor data through wiring on right. Different colors along bottom indicate 4 different sensor areas. 25,50,75, and 100 [mm] spans

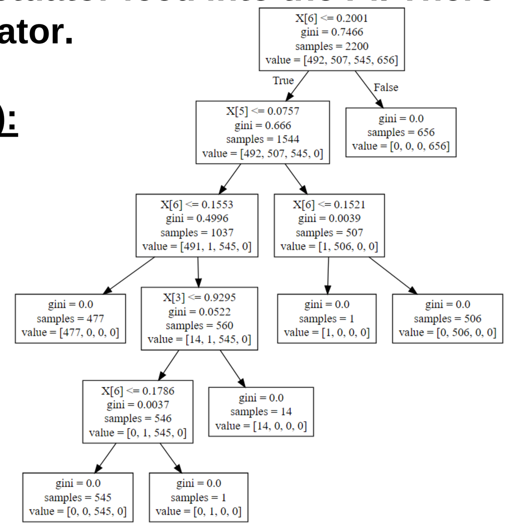

Fig 5: One tree out of random forest classifier.

Machine Learning:

The learning and prediction algorithm is based on Scikit-learn, with data manipulation algorithms and feature engineering. Raw sensor data is pre-processed into events based on pressure and inflation triggers, and all events obtained experimentally are randomized and separated into training and test sets. A random forest classification algorithm of depth 6 (obtained from cross-validation) using a Gini purity measure for cost function. An example tree seen in Fig. 5.

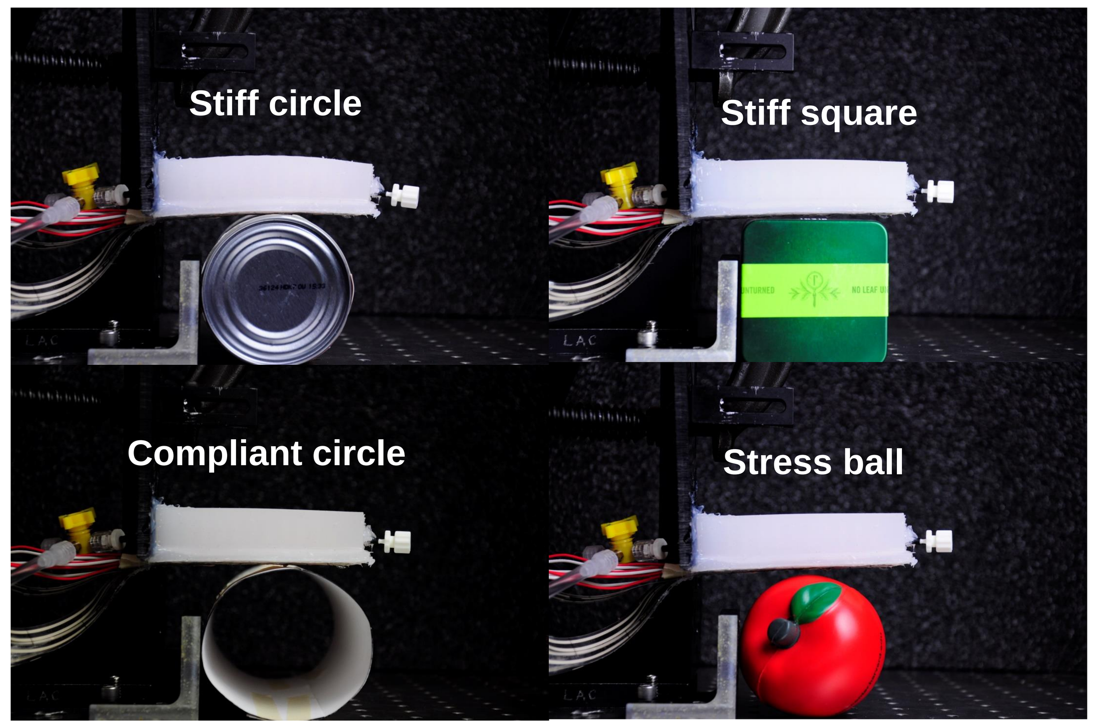

Fig 6: Different objects being grabbed at initial inflation state. All four objects have different cross-sections or different compliance levels. The ML algorithm is able to classify them apart based on real-time sensor data.

Data, Results and Analysis

Data:

- Data is acquired from sensors and stored in Raspberry Pi.
- Each sensor exhibits distinct signal for each object.
- First sensor area (depicted by blue region in Fig. 4) does not provide useful or distinct information.
  - Indicative that we must study optimal placement of sensors.
- Force sensor outputs noisy but contains consistent trend.

Pre-processing Analysis:

- Clean up data and discard uneventful data based on pressure sensors.
- Apply Butterworth signal filter to data and retain 0.05 lowest frequencies of force sensor.
  - This clears out all noise and retains only the useful information.
- Compare co-variance between events for each distinct pressure measurement to see how reproducible each signal is given slightly perturbed input.

Results:

- We have varied the model between Decision Trees, Random Forest, and Neural Networks (NN).
  - With complete data all models have yielded 100% classification accuracy with the exception of NN, which misclassified 1 object in test set.
  - For small number of objects, NN over-fits training data, as a result we drop considering it.
- Using a tree based method, we are able to maintain classification rate using noisy signal or filtered signal.
- Furthermore, when removing training data for an object, we are still able to correctly classify all other objects. Missing training object is classified as a mixture of other objects during accuracy test.
  - This is indicative that signals for different objects are independent of each other.

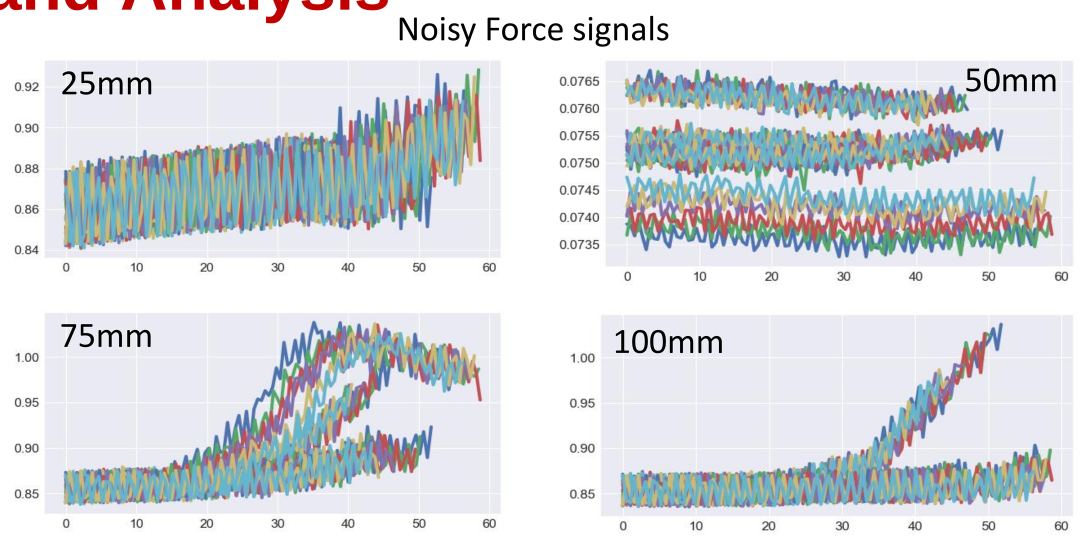

Fig 7: Each subfigure shows separate unfiltered readings from different force sensors. We use noisy data to fit into ML algorithm as training and test. Using this model, test classification accuracy maintains 100%. Vertical axis – voltage reading, horizontal axis - pressure stage.

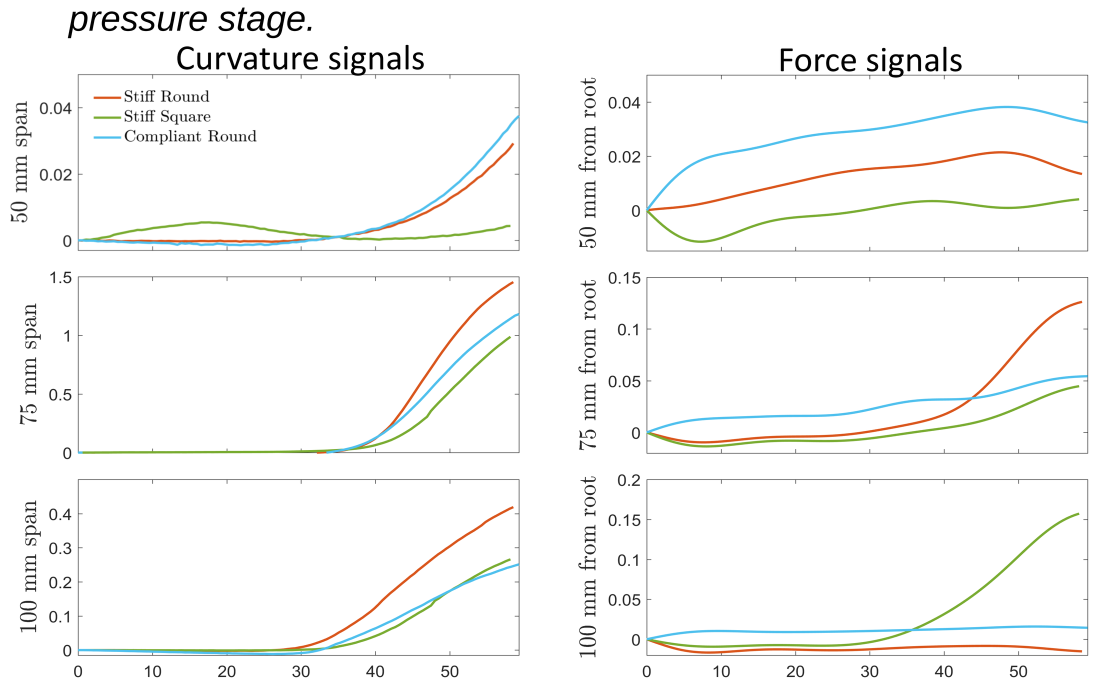

Fig 8: Clean, filtered force data and raw curvature signal. This data is used as the ultimate complete training data for the model. We reduce model variance by factor of 0.5 through using high pass filtered force signal. Vertical axis – voltage reading, horizontal axis - pressure stage.

Conclusion

Classification Accuracy:

- Tree based models maintain 100% test prediction accuracy for relatively small catalogs of objects.
- Model can differentiate between objects with same or similar cross-section but with different stiffness.
- Classification between objects are robust and orthonormal such that when a certain object is removed, classification of other objects remain intact.

Actuator and Sensor Design:

- Depending on objects of interest, placement of sensors need to be optimized to acquire the most useful data.
- Noisy sensor data does not impact classification accuracy on tree-based methods as long as the noise to signal amplitude ratio is relatively small and the noise frequency is much higher then the signal of interest.

Future Studies and Improvements

Machine Learning:

- Given the success of tree based methods thus far we plan to incorporate a better similar method called XGBoost.
- When considering more class objects we must revisit convolutional NN or Deep Learning as a potential algorithm for the model.

Actuator Design

- Improve sensor distribution along the actuator to increase amount of useful data.
- Add more or different types of sensors to the actuator.
- Consider different actuator types with 3D printed embedded sensors.

Beyond Object Classification

- Use parametric equations to infer arbitrary object shapes from sensor data.
- Develop parametric methodology to minimize amount of training data required.

Acknowledgment

- NSF GRFP Grant: DGE-1144152 (MCF, Fellow ID: 2015199385)
- NSF-Harvard MRSEC: DMR-1420570
- NSF BIOMATTER: DMR-1533985
- NSF IMECE NSF Travel Award Grant: CMMI-1649149, Dr. Mary M. Toney

Publications

In preparation. Please contact Matheus Fernandes at m@fer.me for more information.

References

[1] Anadan et al. (2016). Robotic Bin Picking – The Holy Grail in Sight. ***Robotics Online.*** `https://goo.gl/k2GfZt`

[2] Clark and Bhasin (2017). Amazon's Robot War Is Spreading. ***Bloomberg Technology.*** `https://goo.gl/gqY274`

[3] Klingbeil et al. (2011). Grasping with application to an autonomous checkout robot. ***ICRA Conference.*** `https://goo.gl/tFwM9w`

[4] Introduction to XGBoost (2016). ***XGBoost Docs.*** `https://goo.gl/uZ8XCx`