

Online 3D Deformable Object Classification for Mobile Cobot Manipulation

Khang Nguyen, Tuan Dang, Manfred Huber
Learning and Adaptive Robotics Laboratory



THE UNIVERSITY OF TEXAS
AT ARLINGTON

Assistive Mobile Cobots Can Be Used to Grasp Various Everyday Objects for Service Tasks



Courtesy: [Frank, Barbara, et al, 2011](#)



Courtesy: [Google's robotics lab](#)



Image: [Our case](#)

Some Objects Are Easily Deformed Due to Grasping Forces from Robots or from Human Factors



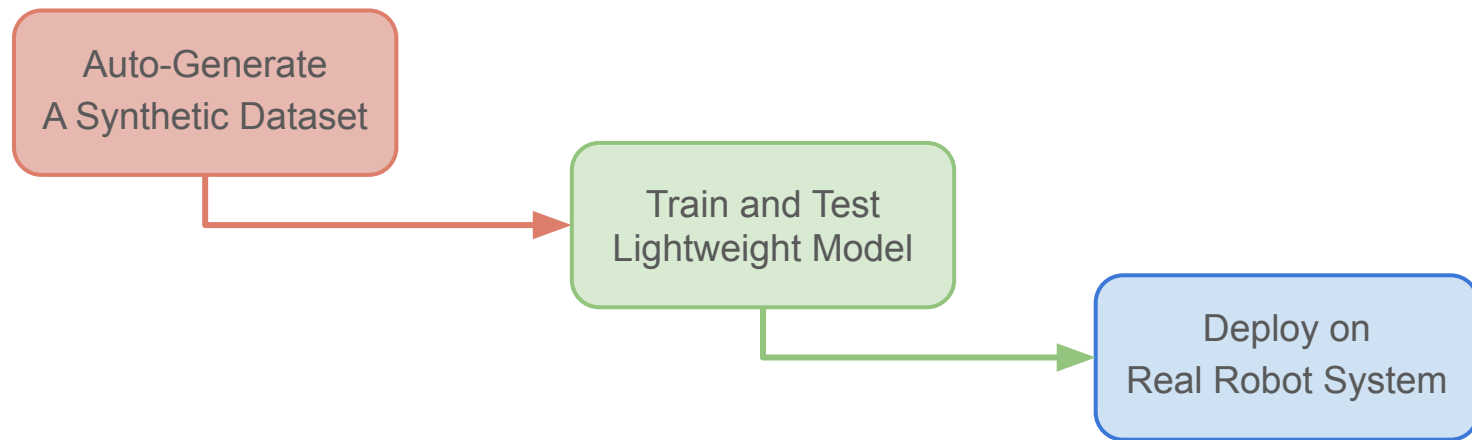
and machine learning (ML) models for vision are highly sensitive to input changes \Rightarrow misclassification problem

3D Deformable Object Classification for Mobile Cobots

Key Contributions:

- Generation pipeline for deformed objects from 3D scan.
- Lightweight ML model to classify objects with deformation artifacts.
- Guarantee online inference for mobile cobots.

Our Pipeline:



Scanning Real-World Deformable Objects



- The MakerBot Digitizer 3D scanner (*right*) is used to scan real-world objects.
- Our raw scans (*left*) include various-sized deformable objects (e.g., tin cans, foam balls, and paper cups).

Deformable Object Generation Procedure

- Downsampling scanned objects
- Identifying graspable region
- Sampling handle points
- Taking handle points for multiple deformations
- Slicing the object for position of opposite handle point
- Defining **inward orientation** and **deformation intensity** for deformation
- Generating deformed meshes



(a) Scan Real-World Object



(b) Downsample Point Cloud



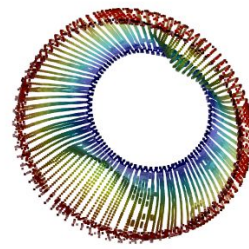
(c) Identify Graspable Region



(d) Sample Handle Points



(e) Slice Point Cloud



(f) Deform based on Handle Points



(g) Generate Deformed Mesh

As-Rigid-As-Possible (ARAP) Deformation with Smooth-Regularization

- Denote a mesh of n vertices as: $\mathcal{M} := (E, V, F)$ with E , V , and F are sets of edges, vertices, and faces in M , respectively.
- We would like a deformed mesh, as follows:

$$\mathcal{M}' := (E', V', F')$$

with E' , V' , and F' are sets of edges, vertices, and faces in M' , respectively.

As-Rigid-As-Possible (ARAP) Deformation with Smooth-Regularization

- Without loss of generality, arrange vertices in the deformed mesh as follows:

$$\begin{cases} \mathbf{v}'_i = \mathbf{c}_i = \mathbf{v}_i, & \text{for } i = 1, 2, \dots, m \\ \mathbf{v}'_i = T_i(\mathbf{v}_i), & \text{for } i = m + 1, m + 2, \dots, n \end{cases}$$

where:

$T_i(\cdot)$ represents the transformation from the original vertex,

\mathbf{v}'_i is the transformed vertex of \mathbf{v}_i , and

\mathbf{c}_i denotes a constraint point in M' that is invariant to vertex \mathbf{v}_i in M .

As-Rigid-As-Possible (ARAP) Deformation with Smooth-Regularization

- The formula for ARAP deformation with smooth regularization is written as:

$$\mathcal{E}(V') = \sum_{i=1}^n \left\| \frac{1}{|\Omega_i|} \sum_{j \in \mathcal{N}(\mathbf{v}'_i)} \omega_{\text{cot}}(\mathbf{v}'_i - \mathbf{v}'_j) - R(\mathbf{v}'_i) \cdot \delta(\mathbf{v}_i) \right\|_2^2 + \sum_{i=m+1}^n \left\| \mathbf{v}'_i - \mathbf{c}_i \right\|_2^2 + \sum_{i=1}^n \sum_{j \in \mathcal{N}(\mathbf{v}'_i)} \alpha_{\text{smooth}} \cdot S(\mathcal{M}) \left\| R(\mathbf{v}'_i) - R(\mathbf{v}'_j) \right\|_2^2,$$

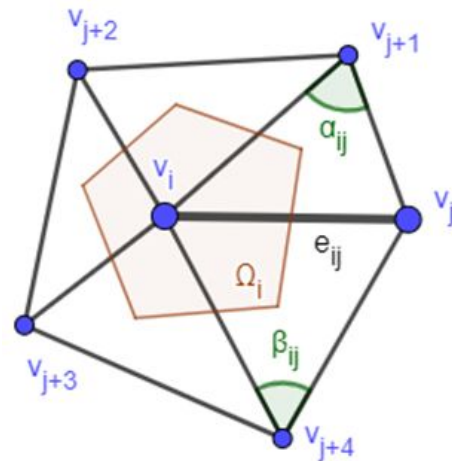
we want to optimize this

differential term

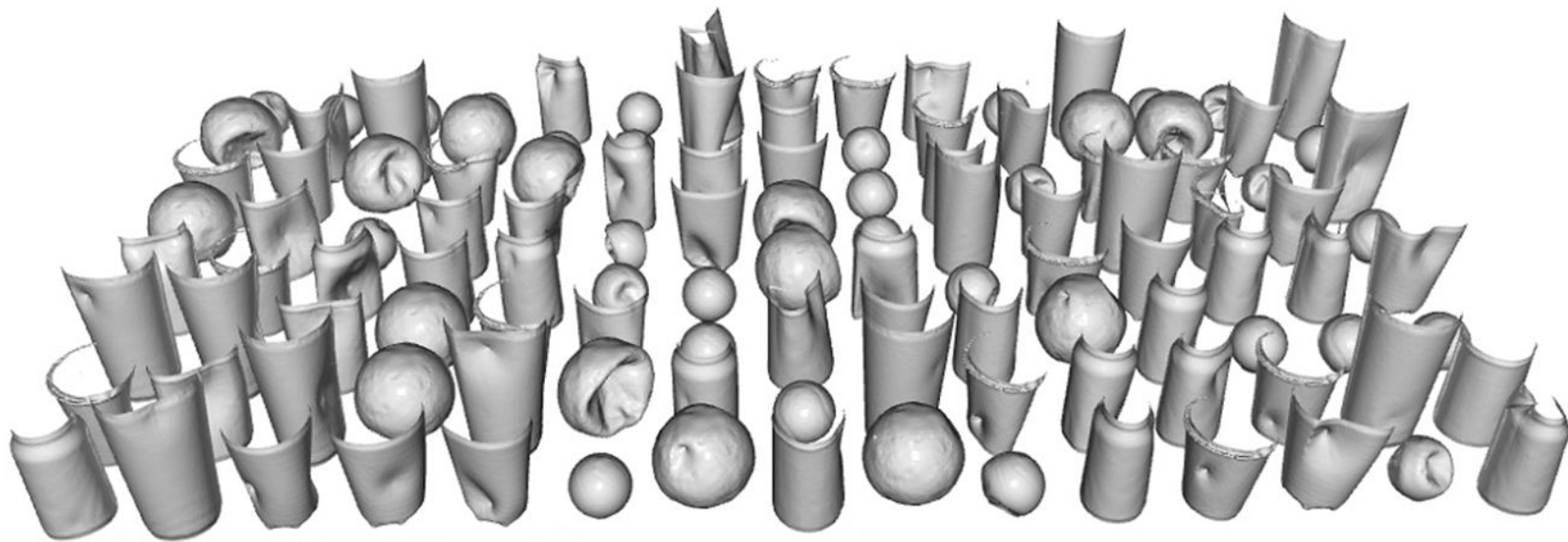
smooth regularization term

$$\text{with } \delta(\mathbf{v}_i) = \frac{1}{|V|} \sum_{i=1}^{|V|} \frac{1}{|\Omega_i|} \sum_{j \in \mathcal{N}(\mathbf{v}_i)} \omega_{\text{cot}}(\mathbf{v}_i - \mathbf{v}_j)$$

$$\text{and } \omega_{\text{cot}} = (\cot \alpha_{ij} + \cot \beta_{ij})/2$$

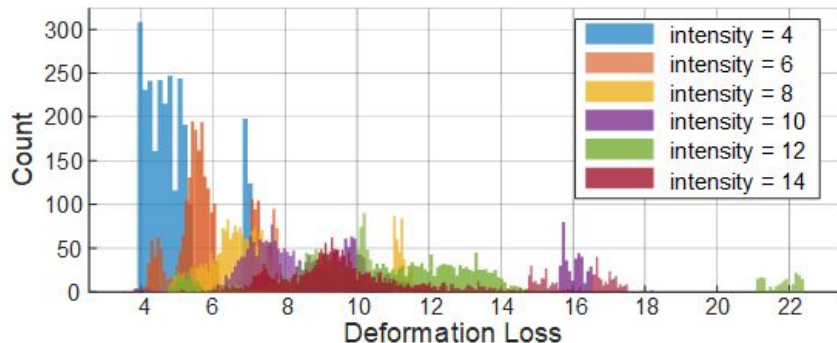


Demonstration of A Subset of the Auto-Generated 3D Deformable Object Dataset



The objects are viewed using Open3D.

How Well the Dataset Distributes In Terms of Deformation Loss?

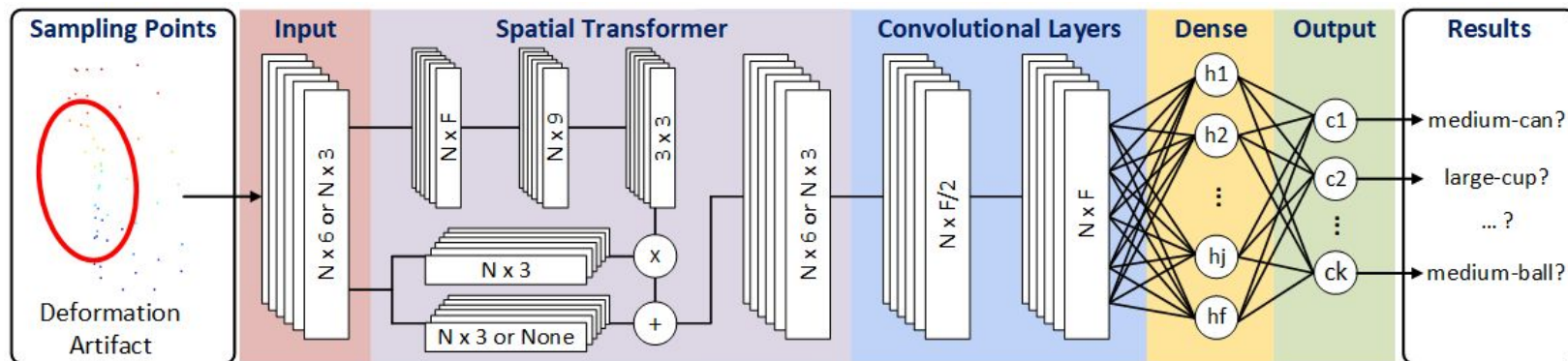


Well-distribute across the domain.

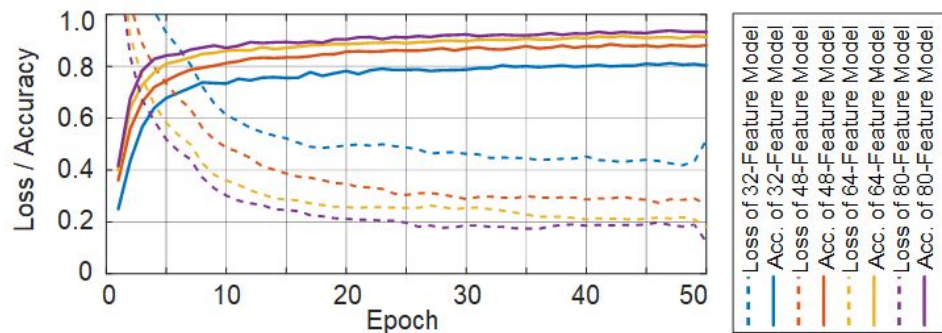
- The deformation loss is computed as weighted losses of:
 - Chamfer distance loss
 - Mesh edge loss
 - Mesh normal loss
 - Laplacian smoothing loss
- **Intensity** represents the inward displacement of handle points during the ARAP deformation.

Classification Network for 3D Deformable Objects

- Takes sampling points with *deformation artifact* as input.
- Passes through a spatial transformer, two convolutional layers, and one fully-connected (dense) layer.
- **Spatial transformer** is used to find the representation that is invariant to *permutations, rotations, and translations* in point clouds.
- Returns probabilities among classes at the output layer.



Training Performances of Designed Networks

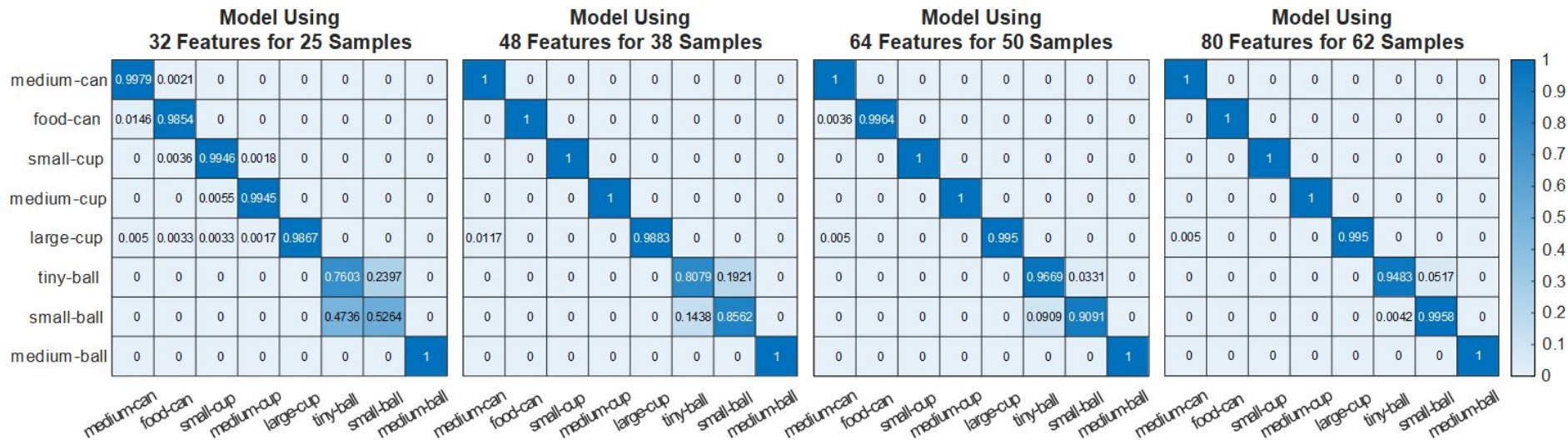


Training performance of deformable object classifier with multiple feature choices

E.g., 64-feature model means the neural network having F equals 64.

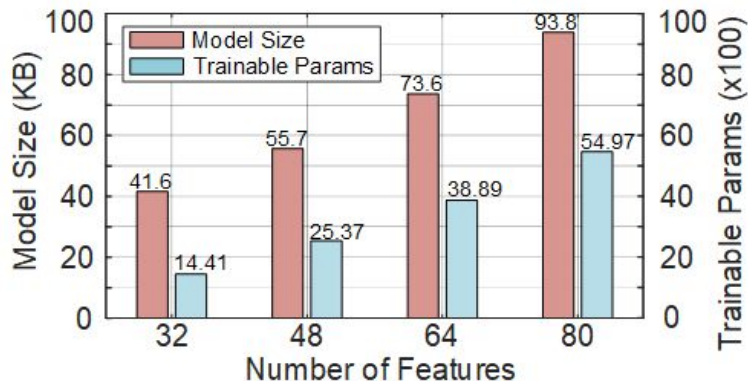
- The training stage converges relatively fast on the NVIDIA RTX 4090 (24 GB) GPU:
 - batch size of 256
 - 50 epochs
 - 10 minutes of training time
 - approx. 90% of accuracy on validation set

Testing Performances of Designed Networks



**The more features the network learned,
the better accuracy it achieves.**

Trainable Parameters and Model Sizes



- The best model from our observation:
 - 80 features are learned
 - about 100 KB
 - about 5,500 trainable parameters

Suitable for on-robot deployment!

GitHub Repository

- **Link:** https://github.com/mkhangg/deformable_cobot

The screenshot shows the GitHub repository page for 'deformable_cobot'. The repository is public and has 2 commits. The file list includes: __pycache__, cache, checkpoint, config, deployment, generator, images, log, .gitattributes, DNet.py, README.md, classify_vino.py, cropping.py, dataset.py, eval.py, export.py, net_utils.py, normalizing_cropped_pcd.py, and plot_data.py. The repository is described as '[ISR '23] Online 3D Deformable Object Classification for Mobile Cobot Manipulation'. It has 0 stars, 1 watching, and 0 forks. The repository is created on Jun 23, 2023.

The screenshot shows the README.md file for the 'deformable_cobot' repository. The title is '[ISR '23] Online 3D Deformable Object Classification for Mobile Cobot Manipulation'. The authors are listed as 1. Khang Nguyen, 2. Tuan Dang, and 3. Manfred Huber. The abstract states: 'Vision-based object manipulation in assistive mobile cobots essentially relies on classifying the target objects based on their 3D shapes and features, whether they are deformed or not. In this work, we present an auto-generated dataset of deformed objects specific for assistive mobile cobot manipulation by utilizing intuitive Laplacian-based mesh deformation. We first determine the graspable region of the robot hand on the given object's mesh. Then, we uniformly sample handle points within the graspable region and perform deformation with multiple handle points based on the robot gripper configuration. In each deformation, we also identify the orientation of handle points and prevent self-intersection to guarantee the object's physical meaning when multiple handle points are simultaneously applied to the object's mesh at different deformation intensities. Finally, we test our generated dataset on the Baxter robot with two 7-DOF arms, an integrated RGB-D camera, and a 3D deformable object classifier. The result shows that the robot is able to classify real-world deformed objects from point clouds captured at multiple views by the RGB-D camera. The demo is available at YouTube.' The image shows a red mobile robot arm (Baxter) interacting with a 3D model of a deformable object on a screen.

Demonstration Video



Click to see the demo video.

Thank you for listening!

