

# Textureless Object Recognition: An Edge-based Approach

1<sup>st</sup> Frincy Clement

*dept. of Computing Science  
University of Alberta  
Edmonton, Canada  
frincy@ualberta.ca*

2<sup>nd</sup> Kirtan Shah

*dept. of Computing Science  
University of Alberta  
Edmonton, Canada  
knshah@ualberta.ca*

3<sup>rd</sup> Dhara Pancholi

*dept. of Computing Science  
University of Alberta  
Edmonton, Canada  
dhara@ualberta.ca*

4<sup>th</sup> Gabriel Lugo Bustillo

*dept. of Computing Science  
University of Alberta  
Edmonton, Canada  
lugobust@ualberta.ca*

5<sup>th</sup> Dr. Irene Cheng

*dept. of Computing Science  
University of Alberta  
Edmonton, Canada  
locheng@ualberta.ca*

**Abstract**—Textureless object recognition has become a significant task in Computer Vision with the advent of Robotics and its applications in manufacturing sector. It has been challenging to obtain good accuracy in real time because of its lack of discriminative features and reflectance properties which makes the techniques for textured object recognition insufficient for textureless objects. A lot of work has been done in the last 20 years, especially in the recent 5 years after the TLess and other textureless dataset were introduced. In this project, by applying image processing techniques we created a robust augmented dataset from initial imbalanced smaller dataset. We extracted edge features, feature combinations and RGB images enhanced with feature/feature combinations to create 15 datasets, each with a size of 340,000. We then trained four classifiers on these 15 datasets to arrive at a conclusion as to which dataset performs the best overall and whether edge features are important for textureless objects. Based on our experiments and analysis, RGB images enhanced with combination of 3 edge features performed the best compared to all others. Model performance on dataset with HED edges performed comparatively better than other edge detectors like Canny or Prewitt.

**Index Terms**—Computer Vision, Textureless object detection, Textureless object recognition, Feature-based, Edge detection, Deep Learning

## I. INTRODUCTION

In recent years, a lot of labor-intensive activities in manufacturing sector has been automated using robots. In industrial assembly line, a robotic arm is used to pick up small objects and place it in its location. These industrial objects are mostly textureless and are different from textured objects in terms of lack of discriminative features. Often, they are smaller in size and can be similar to each other. In real world, they are present along with multiple objects of different classes in different scene configurations, which makes the recognition even more challenging.

In the early 2000s, detection and recognition of the textureless objects seemed to be a daunting issue, even for the state-of-art detectors at that time, like SIFT (Scale Invariant

Feature Transform) and SURF (Speeded UP Robust Features). This was due to their dependence on the high-featured, texture-rich informative regions which were sparse in the textureless occurrences. Ultimately, this led to several works which contributed for the detection and recognition of textureless objects, which can be thus classified into three categories: View-based, Feature-based, and Shape-based.

### A. View-based

View-based methods work by comparing the object of interest (OOI) and the pre-calculated 2D views of that object, denoted by aspect graphs. This method can provide a rough estimate of the pose of the object, but it suffers from computational load of searching a large spatial domain. To overcome this issue, Cyr and Kimia [3] proposed a method that grouped similar OOI together under a single class and similarity scores were calculated to perform recognition. Eggert et al. [1], Ulrich et al. [8] and Steger et al. [2] introduced various improvements on the aspect graphs and similarity metric. This approach, however was not widely used because of its high complexity.

### B. Feature based method

Tombari et al. [9] devised a method called BOLD in which they attempt to tackle the issue of object detection under clutter. In this method, neighbouring line segments were aggregated to form a feature representation of the object. These line segments are invariant to rotation, translation as well as scaling. A limitation of BOLD is that it fails if there is clutter in the line segment selected for feature description.

Intended for object recognition and tracking using an RGB-D sensor, Jiang et al. [17] proposed a method that detects and recognises small objects. The image is divided into various parts using image segmentation. Then it is sent to a classifier as an input which classifies the object using the features extracted in the earlier stages.

Hodan et al. [13] proposed a technique for detecting textureless objects using AR, which used an edge-based detector which gave better results than canny edge detector. And efficient towards the object detection as it searched edgelet in stripes instead of wedges.

While the above-mentioned methods do a commendable job of detecting textureless objects, they were computationally expensive. In 2016 Jacob et al. [18] introduced BORDER which is an acronym for Bounded Oriented Rectangle Descriptors for Enclosed Region, in which they use a rectangle to encapsulate the object so that minimum outliers are detected. It outperformed the previous state-of-the-art descriptors BOLD and Line2D when compared for images having occlusion and clutter.

Counting the number of textureless objects present in a scene was still a challenge until Verma et al. [20] proposed a method which counts the objects using shape and colour feature which are obtained through morphological boundary extraction and segmentation. The uniqueness in this method was it was invariant to scaling and rotation. Further, different models such as SVM, KNN etc were used for the classification.

To improve accuracy, Thoduka et al. [22] in 2016 came up with an approach which used RGB-D based features to recognize objects in the industrial assembly line such as nuts, bolts, bearing large and small black profiles etc. The approach performed efficiently on most of the objects in dataset but they had poor performance recognizing smaller objects.

Following upon his scholarly work, Hodan et al. [24] improved upon his previous research by presenting a new dataset in 2017, “T-less” publicly for better understanding of 3D object from different dimensions. This dataset contains industrial objects which have no texture while having feature similarities of shape/size which makes it quite hard to differentiate.

To overcome the short comings of previous methods, BIND [23] was introduced during 2017. It is a detector that uses binary nets having multiple layer for accurate detection of textureless object. To improve the speed of object detection, in 2018 Hancheng et al. [36] introduced a faster approach to detect textureless objects even in the cluttered background and transformations.

Fang et al. [32] deduced that for object recognition, the background of an image affects the performance of neural networks, and hence came up with a method called DOG, built upon CNN which effectively removed the background of an input image for improving the efficiency of the classifier.

### C. Shape Based Approach

Template matching is a popular method in textureless object recognition, dating back as early as the 2000s in lieu of the fact that it provides high mAP and close to real-time detection. But, up until now, templates were generated using one modality only (depth or colour or point cloud or 2D). However, in 2011 Hinterstoesser et al. [6] put forth the idea of multi-modal template matching. The template comprises of both

the image cue (colour information) as well as depth cue (3D) information. The method can detect objects in real-time as well as under heavy clutter. A caveat of the above approach is that it is not invariant to translation, rotation, or scaling.

In the following year, Hinterstoesser et al. [7] overcame the afore-mentioned issue by considering all the gradient orientations for a template instead of just the dominant ones. Due to this, the method was more robust to translations. While it does suppress the caveat of the above approach to a certain extent, it is still not invariant to scaling.

In 2015 Hodan et al. [13] came up with a very efficient way of template matching, resolving a lot of its computational complexity in performing searches of each sliding window with each stored template. This method resulted in real time performance with low false rates and multiple object detection.

In 2018, an edge-based Hierarchical Template Matching algorithm was proposed by Tsai et al. [28] based upon Line2D algorithm, which gave an efficient method for real-time object detection and recognition. As per this paper, edges are identified as the most stable feature for a textureless object. This was the baseline information for building our project.

2018 saw many approaches for textureless object detection. PoseCNN (current state-of-the-art), an approach by Xiang et al., [27] is a convolutional network for predicting object pose with varied orientations. It predicts object's translation matrix by means of calculating object's distance with respect to the camera. They also calculate the object's rotation matrix. Midway through 2019, Park et al. [34] proposed a method that takes depth image as an input and a region of interest (ROI) is generated.

In the next section, we introduce the existing work done in the field of textureless object recognition by using a combination of image processing and deep learning techniques. Our proposed method is explained step by step in Section III. Furthermore, Section IV details the experiments performed, results and analysis. The report ends with Conclusion and future work, Acknowledgement, Author Biography and References.

## II. RELATED WORK

Our proposed work uses image processing techniques combined with machine learning techniques for textureless object recognition. We perform image processing techniques to extract features and create multiple training sets. We compare these datasets using machine learning techniques. In this section, previous researches which uses similar approach are summarized.

Jiang et al. [18] proposed an RGB-D based method for the detection of various types of common objects. They used depth filtering alongside clustering, coupled with the watershed algorithm on the depth data for the target object recognition. Once recognized, they apply a tracking method to reduce the search space and the computational load. Further, the method was evaluated using a random forest classifier. However, for real-time object detection, only RGB dataset was used.

Wang et al. [19] proposed a method for object recognition using model-based CNN learning. They introduced a unified

structure composed of automatic object reconstruction and multitask learning aimed at doing object recognition by model-based training alone. This method reduced the gap between real images and synthetically rendered images for better accuracy.

Later, a technique to count textureless objects with shape and colour feature was developed by Verma et al. [20]. First, the shape and color features were extracted during pre-processing using morphological boundary extraction and segmentation via mean hue value (colour features) and Hu-moments (shape features). Next, SVM, kNN, NN and tree bagging were applied for the classification of objects. Through experiments, they observed that tree bagging has the best accuracy of all the classifiers. Finally, the classified objects were counted by drawing bounding boxes around them.

Thoduka et al. [22] put forth a method for object detection using 3D point cloud. They defined size, shape, and color of objects by using various metrics including the bounding box. The extracted features were then passed through SVM for classification. However, the accuracy of the method is inversely proportional to the size of the object in each image.

A breakthrough in the field of textureless object recognition happened when Hodan et al. [24] introduced T-less dataset for 6D pose estimation. The dataset houses thirty industry-relevant textureless objects with no discriminative colour or illumination properties. They are not symmetrical with respect to shape or size. It contains 39K training images along with 10K testing images that exhibit occlusion and clutter. It also has manually created 3D CAD model, as well as one which is constructed semi-automatically. For our project, we used the 2D images taken from 3D printed models of this dataset as our baseline data.

Hodan et al. [13] came up with a method using purely edge detector which uses edgelet strips in place of wedges. When the binary map of edges is provided, this method takes the edglets of equal fix lengths and they are represented by their midpoints and orientation, which helps in tracking the orientation. Two images are matched using their distance transformation in orientation which proves the presence of that object in image even if more than one same object is present.

A technique called PoseCNN introduced by Xiang et al. [26] used for 6D pose estimation, estimated the 3D translation and rotation of an object by predicting the center of the object's distance from the camera. For this process, they also introduced a novel loss function called P loss for objection detection with clutter. Speaking of which, clutter and background are important aspects when it comes to image segmentation. Fang et al. [33] devised a method called DOG which removes the background of the images to improve the object detection accuracy. The method uses DCGAN for classification.

Tsai et al. [28] came up with a new approach for Template matching which proved to be an efficient method for object detection and recognition. They chose edges as they were identified as the most stable feature for a textureless object. After image enhancement, the input image goes through the edge-based template matching algorithm. It detects the color

edge by finding the one with maximum magnitude across all color channels. These detected edges are then quantized into one of 8 bins and one-hot encoded for computational efficiency. The one-hot encoded edge orientations are then spread using dilation. The algorithm then checks for similarity with the precomputed database which has edge templates of reference images and lookup tables of edge orientations, which will help in efficient matching and calculation of similarity score.

For the detection of textureless objects, the importance of depth images over colour images cannot be overstated. Park et al. [34] developed a method called Multi-task template matching for object detection, segmentation and pose estimation using depth images. They put forth a method to match the Nearest Neighbour template using a similarity score. Since shared feature maps were used to predict the segmentation mask and pose of an object, additional training was not required for a new object. For real-time object detection and recognition, Region of Interest (ROI) predicted by ResNet-50 architecture pretrained offline on the templates, were compared with the ROI of test image predicted real-time by another ResNet-50.

### III. PROPOSED METHOD

Our work focuses on recognition of textureless objects in 30 classes. In this section, we will discuss how we acquired the intial dataset, what image processing techniques are applied on the input images to convert it into features and feature-boosted versions of original images and which classifiers are selected for object recognition. The “Fig. 1” depicts the block diagram of our proposed method.

#### Project Implementation Structure

Our project was implemented in 3 parts. In Part 1, we focused on applying image processing techniques to create a balanced augmented dataset from ground truth. In Part 2, we apply edge operators to create 14 new datasets to perform feature comparison. In Part 3, we train a classifier and predict results.

##### *Part 1: Creation of Initial Augmented Data*

- Image Data Acquisition
- Obtaining Ground Truth data
- Data Balancing and Augmentation
  - Manual: Image Processing Techniques
  - Automatic: Augmentor API

##### *Part 2: Creation of 14 new datasets for feature comparison*

- Creation of Feature Only Dataset
- Creation of Feature Combinations
- Creation of Feature Enhanced RGB Dataset
- Creation of Feature Combination Enhanced RGB Dataset

##### *Part 3: Training and Testing on Multiple Classifiers*

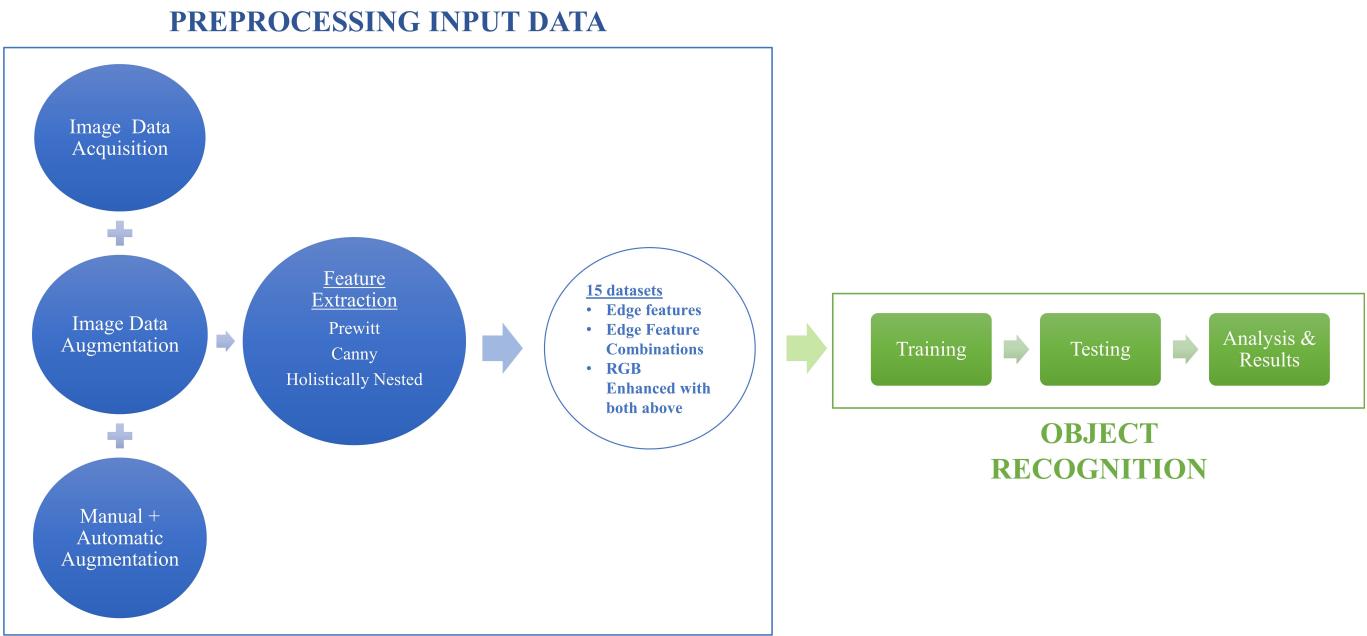


Fig. 1. Project Workflow

## Part 1: Creation of Initial Augmented Data

1) *Image Data Acquisition*: The dataset we used for our experiments are based on TLess Dataset [24] introduced by Hodan et al. 30 textureless objects were 3-D printed from the CAD model and using an Intel RealSense camera, 3D images were captured at the Multimedia Research Lab, University of Alberta. We used the 3D-rendered 2D images to start our implementation for object recognition.

2) *Obtaining Ground Truth Data*: We started our experiments with 27000 images that contained textureless objects from 30 different classes. The objects that were captured covered only small portion of the images. We used regionprops() from sci-kit learn library to obtain bounding boxes for each object. From those bounding boxes, the objects were cropped leaving few pixels on all sides. The bounding box information was also saved as annotations for training any of YOLO versions in the form: Classname, x-center, y-center, height, width – all normalized to image height and width. The different sample images captured from Intel RealSense camera and the output of regionprops() are shown in “Fig. 2”.

3) *Data Balancing & Augmentation*: The data provided was captured in 4 different orientations for most of the classes except for the ones that had unique features. The data was unevenly distributed among 30 different classes. As seen on “Fig. 3” Whereas most of the classes had ~1200 images, some had only ~300.

To balance the dataset, we calculated a balancing factor. While applying manual augmentation, we considered this balancing factor to remove the imbalance in dataset.

The number of images captured using Intel RealSense is 1000 per object, which is too small to efficiently train a

classifier. Hence, we need to apply image augmentation to generate artificial data. We augmented the input image in two ways: Manually applying image processing techniques and using Augmentor API.

a) *Manual Augmentation*: As proposed in the recognition method by Huang et al. [30], we applied the following techniques on the ground truth data with a balancing factor. The results are shown in “Fig. 4”

- 1) Contrast Enhancement – to improve the contrast of the textureless object with respect to the background
- 2) Addition of Noise – to avoid overfitting of input data
- 3) Brightness transformation – to simulate the object detection in the lighting in manufacturing facility
- 4) Addition of Blur – to account for the blurring that can be caused by the camera movement

b) *Automatic Augmentation* by using an API: Augmentor [26] by Bloice et al. is a software package which provides high level API in Python and Julia with a pipeline-based approach to perform various operations like rotation, translation, scaling, cropping, shearing etc. We used this package to augment our dataset to add following:

- 1) Rotation
- 2) Random Erasing (80 % intact)
- 3) Random cropping (80 % intact)
- 4) Flip (Left-Right & Top-Bottom)
- 5) Skew (Magnitude – 0.3)

A sample of the results of automatic augmentation are shown in “Fig. 5”.

Upon performing manual and automatic augmentation, we obtained a balanced dataset that contains similar number of

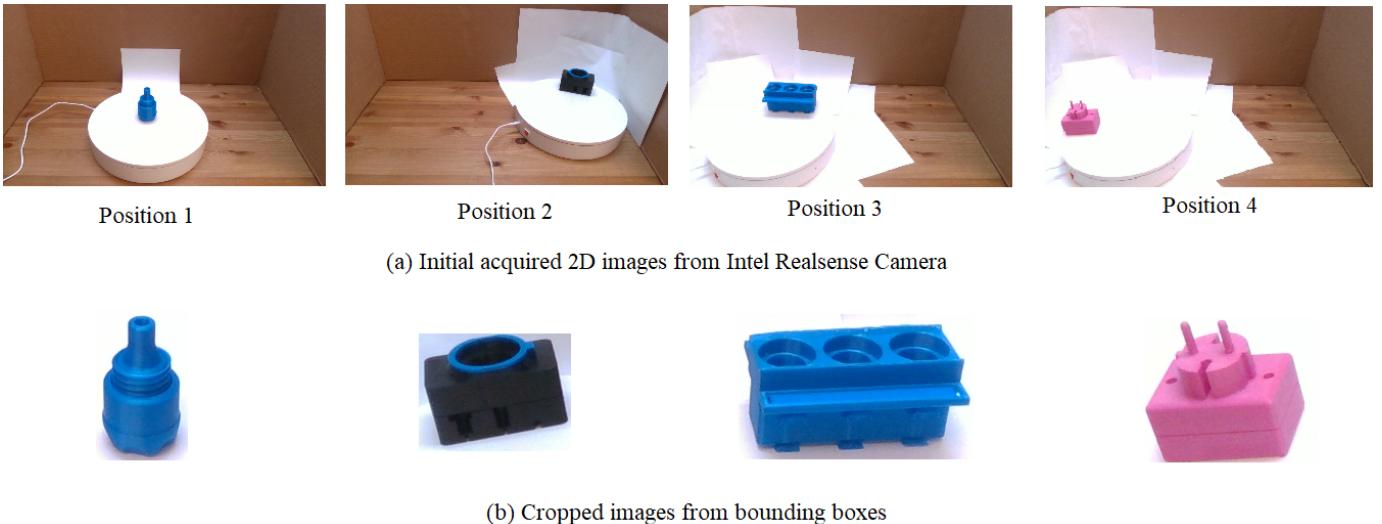


Fig. 2. Data Preprocessing to obtain ground truth objects from initial acquired 2D images

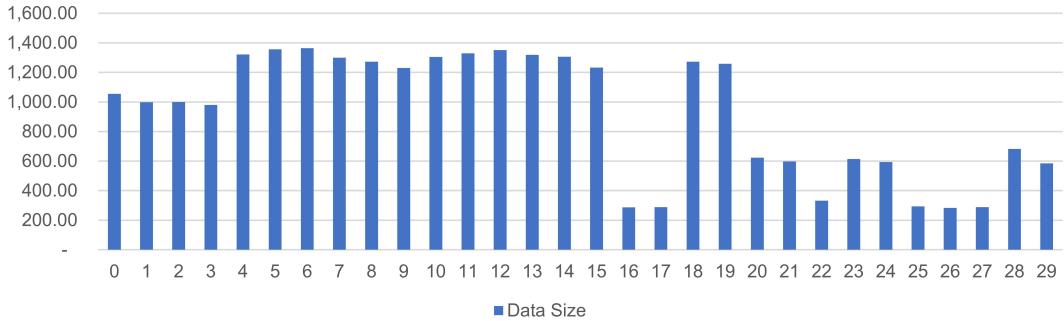


Fig. 3. Distribution of data across 30 classes

images for all the classes. The new dataset comprised of 340,000 images belonging to all 30 classes.

## Part 2: Creation of 14 new datasets for feature comparison

From literature review, it could be inferred that for a textureless object, edges are the most important feature. In part 1, we augmented the data from 27,000 (Ground truth) to 340,000 (Augmented data). For obtaining the edge features, we implemented three different edge detectors on the ground truth data to obtain the corresponding edge features.

- 1) Canny Edge detector: By applying non-max suppression on the image gradient intensities, and applying double threshold, it finds the strong edges by suppressing the weak ones.
- 2) Holistically Nested edge detector (HED): Xie et al. [37] developed a neural network, which uses a deep neural network to automatically learn hierarchical edge maps thereby identifying the edges of the objects in images.
- 3) Prewitt operator: By convolving with a 3x3 kernel with the input image, it calculates the gradient of image inten-

sity at each point by combining vertical and horizontal derivative approximations.

Canny edge detector thin dotted lines along the edges of the object, but it was not continuous. HED provided edges of an object which gave thick outlines around the object. Prewitt gave thick outline for the object and thin curvy lines like canny for the inside part of the object. Thus, we created 3 datasets of 340,000 images containing Canny features, HED features and Prewitt features. We also considered Sobel Edge Detector but since it gave a similar result as Canny, it was not implemented.

The next step was to combine the said features for all permutations to observe if they can perform better. First, we combined Canny (thin curvy outlines) with HED (thick outlines) which yielded us good results in the form of well-defined clear outlines of the edges. We then combined HED (thick outlines) with Prewitt (thin curvy outlines) which produced good results, but the outlines were not as well-defined as Canny-HED, especially the inner edges. For the third experiment, we implemented a combination of Canny (thin curvy lines) and Prewitt (thin curvy lines) which gave us thin outlines as well

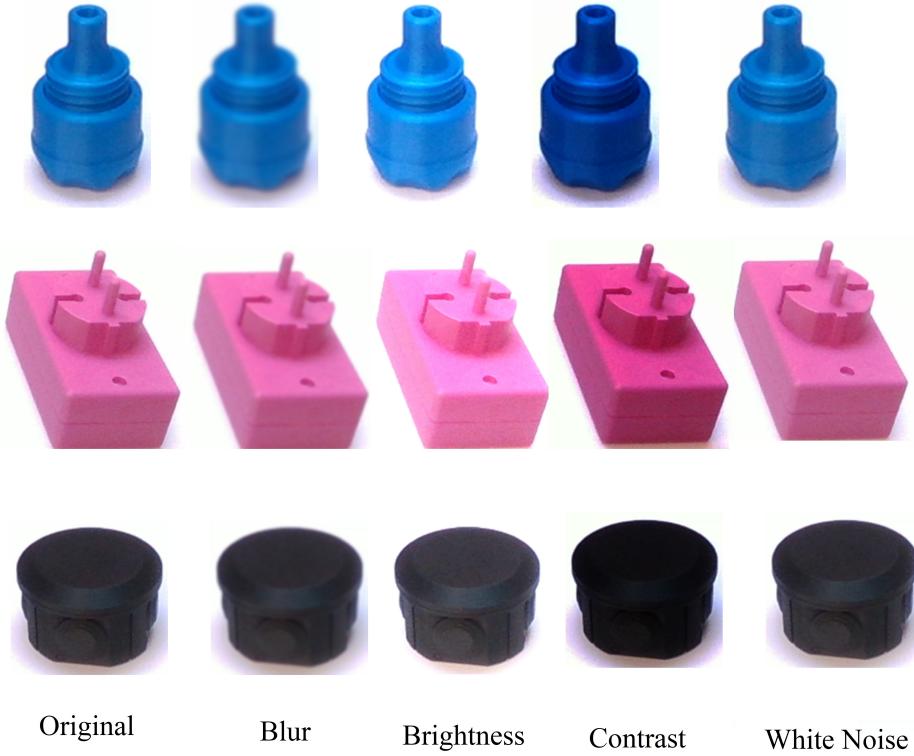


Fig. 4. Results of Manual Augmentation

as inner lines of the object. Finally, the combination all three edge detectors – Canny, HED and Prewitt which yielded us thicker and more prominent edges for the object. So, we now had a total of 7 datasets, each containing 340,000 images of features and feature combinations. A sample from these 7 datasets are shown in “Fig. 6”.

To retain the original RGB channel features, we performed overlay of above edges and edge combinations on the original RGB images to get edge enhanced RGB images as seen on “Fig. 7”.

Hence, in total we created 15 datasets containing 340,000 images each to amass a total of 5 million images along with bounding boxes and their annotations.

### Part 3: Training and Testing on Multiple Classifiers

Our research is focused on understanding the significance of edge features for textureless objects and to see if edges or edge enhanced pixel intensities can help improve recognition results. Hence, we planned to train our datasets on multiple models for comparing its performance and to have an unbiased result irrespective of choice of model.

*1) Model Selection* Different models were considered based on the following criteria:

- Support incremental learning as we have large dataset (15 GB per dataset)
- Faster training (45 minutes per dataset), as we had 15 datasets to train and predict

- Follows similar input/output dimensions for training
- Have similarity in input pre-processing
- Can be implemented efficiently inside a loop without duplicating lines of codes

With the above selection criteria in mind, we selected to train 4 models from sci-kit learn library:

- 1) Stochastic Gradient Descent (SGD): It helps implement linear SVM or Logistic Regression based on the loss type selected, on a large scale. It performs really well especially when the data is sparse, which is the case of our dataset, as we are comparing edge features, which will eventually result in a nxn sparse matrix with only edge pixel locations having ‘1’ and all others ‘0’. We used this model with loss type ‘log’ for Logistic Regression.
- 2) Perceptron: Perceptron is a simple classification algorithm which is faster than SGD, especially with ‘hinge’ loss and results in comparatively less sparser models. It does not require a learning rate and is not regularized; it learns from its own mistakes. Hence, it is easier to perform hyperparameter tuning with this model.
- 3) Passive-Aggressive Classifier with hinge loss: Passive-Aggressive algorithms are used mainly for large scale learning. It does not need learning rate, but has regularization parameter. We have tried the classifier on both hinge loss and squared hinge loss as it would give us an understanding on what works better for our data

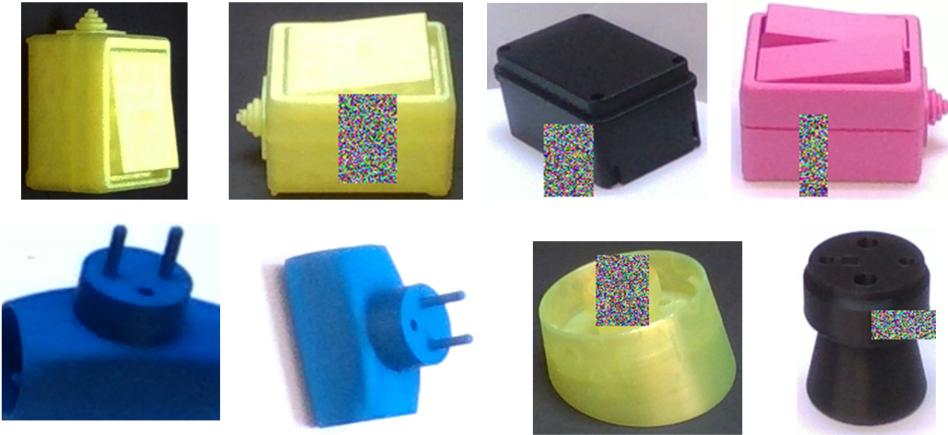


Fig. 5. Results of Automatic Augmentation using Augmentor API

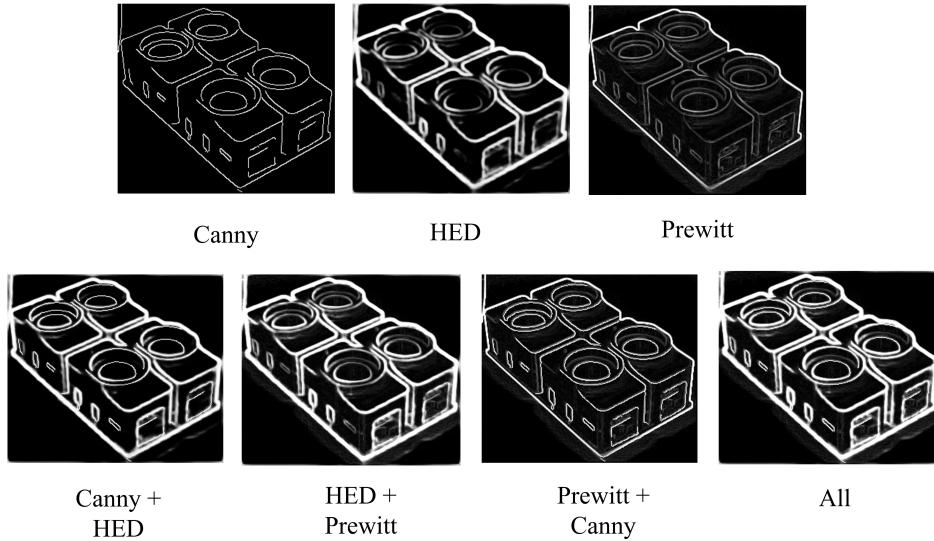


Fig. 6. Sample of Features Only and Feature Combination dataset

#### 4) Passive-Aggressive Classifier with squared-hinge loss

#### 2) Data Preprocessing

We have created 15 datasets: 7 datasets with edges and edge combinations, which are binary images and 7 datasets with edge-enhanced RGB images. Before sending the inputs to the model we have performed following preprocessing steps:

- 1) Created class labels for each input image
- 2) Created feature matrix for input images as a numpy array, with each row corresponding to each input image
  - a) For edges only inputs, we did not apply any initial image processing, but for edge enhanced RGB images, we initially converted them to grayscale
  - b) The result is then resized to 200x200 to have match the dimensions for all input images

c) The resized input is then flattened from 200X200 to 40000X1 array

3) From step 1 and 2, we created each dataset with X as input feature matrix of size 340000 X 40000 x 1 and y as output labels with size 340000 X 1 x 1

#### 3) Fitting and Training the model in batches

Once we have the input dataset, we keep 20% data aside as unseen data for final testing. The remaining 80% is then passed to the model Pipeline as batches which gets divided into 75% and 25% to be used for training and validation.

A batch of 5000 inputs are passed in every iteration into the Pipeline, which has a StandardScaler() and Model (). The StandardScaler() helps to standardize the input features and normalize the values to be between 0 and 1. The standardized

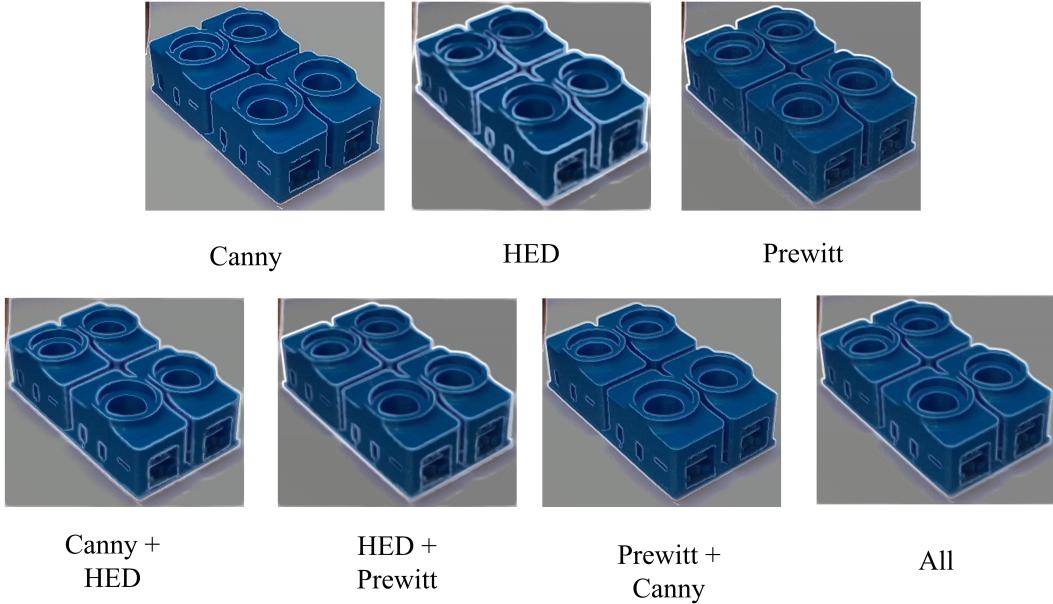


Fig. 7. Sample of RGB images enhanced with Features Only and Feature Combination

features are then passed to each model for training and validation. The progressive training and validation accuracies are noted and compared. The final model performance is noted and compared for Accuracy in “ 8” and F1 score (balance between precision and recall) in “ 9”. To understand if the model is overfitted on any specific dataset, we have captured the difference between average of progressive testing and validation accuracies in “ 10” .

#### *Analysis and Result on Model Performance*

- 1) The best performance across all datasets for Accuracy as well as F1 score was presented by 15th dataset which is the one with RGB images enhanced with all edges combined
- 2) Features only and its combination dataset has shown considerably higher accuracy and F1 score for all classifiers compared to feature enhanced RGB versions. But it is to be noted that as per table 3, features only dataset has overfitted the model compared to its feature enhanced version, because of the high level of sparseness in the input feature matrix.
- 3) Across all the three edge detectors used for comparison, edges features using HED technique performed the best as a single feature and in combination whereas Canny scored the lowest scores

#### *4) Predicting on test dataset*

We have trained 4 models on 15 datasets, which is a total of 60 trained models. We test all of them on two different datasets: 1) Unseen images which are similar to the training dataset 2) Random images taken using a phone camera with

different backgrounds

##### *a. Performance on Unseen Dataset with White background*

As seen on “ 11” the model performance on the dataset with white background was similar to what was expected from the model performance during training.

- 1) The best performance across all datasets for accuracy was presented by 15th dataset which is the one with RGB images enhanced with all edges combined
- 2) Features only dataset has shown considerably higher accuracy for all classifiers compared to Feature enhanced RGB versions.
- 3) Unlike training model performance, all features performed almost equally well for features only dataset

##### *b. Performance on random dataset with different background*

As seen on “ 13” the model performance on the random dataset with different background was different from test set 1.

- 1) The best performance across all datasets for accuracy was presented by 15th dataset which is the one with RGB images enhanced with all edges combined.
- 2) Even though there was an improvement in accuracy from initial augmented data to feature enhanced images, the overall accuracy was way lower for test set 2 compared to test set 1. The reason for this could be that since the model is trained on the images with white background, it may not be performing well for the ones with different background. The model only understands the pixels where there are edges. For a test image with background, its edges will also appear in the feature matrix, which makes it difficult for model to understand and predict.

	<b>Dataset</b>	<b>Classifiers</b>				
		<b>SGD</b>	<b>Perceptron</b>	<b>Passive Aggressive I</b>	<b>Passive Aggressive II</b>	<b>Avg.</b>
1	Initial Augmented training data	69.42%	65.78%	56.00%	75.62%	66.71%
2	Features only using Canny	80.50%	80.32%	82.28%	77.24%	<b>80.09%</b>
3	Features only using Prewitt	81.38%	80.30%	80.40%	82.80%	<b>81.22%</b>
4	Features only using HED	83.48%	81.02%	83.70%	82.30%	<b>82.63%</b>
5	Feature Combination - Canny+HED	82.22%	82.94%	82.84%	82.48%	<b>82.62%</b>
6	Feature Combination - HED+Prewitt	82.12%	80.48%	83.58%	82.20%	<b>82.10%</b>
7	Feature Combination - Prewitt+Canny	79.46%	80.24%	81.28%	82.60%	<b>80.90%</b>
8	Feature Combination HED+Prewitt+Canny	-	80.58%	81.94%	83.46%	82.46%
9	RGB enhanced with Canny edges	69.12%	65.30%	78.04%	74.06%	71.63%
10	RGB enhanced with Prewitt edges	72.70%	75.48%	81.20%	77.58%	76.74%
11	RGB enhanced with HED edges	79.08%	82.16%	83.16%	79.72%	<b>81.03%</b>
12	RGB enhanced with Canny+HED edges	65.60%	75.74%	79.48%	73.08%	73.48%
13	RGB enhanced with HED+Prewitt edges	78.66%	78.76%	81.50%	81.64%	<b>80.14%</b>
14	RGB enhanced with Prewitt+Canny edges	75.28%	81.08%	81.36%	75.50%	78.31%
15	RGB enhanced with All edges combined	83.10%	82.58%	81.40%	83.90%	<b>82.75%</b>

Fig. 8. Comparison of Accuracy for 15 datasets across 4 classifiers

	<b>Dataset</b>	<b>Classifiers</b>				
		<b>SGD</b>	<b>Perceptron</b>	<b>Passive Aggressive I</b>	<b>Passive Aggressive II</b>	<b>Avg.</b>
1	Initial Augmented training data	72.22%	67.53%	59.36%	77.30%	69.10%
2	Features only using Canny	81.16%	80.90%	83.23%	78.89%	<b>81.05%</b>
3	Features only using Prewitt	82.25%	81.14%	82.57%	84.19%	<b>82.54%</b>
4	Features only using HED	84.08%	81.84%	84.33%	83.73%	<b>83.50%</b>
5	Feature Combination - Canny+HED	82.83%	83.48%	83.39%	83.18%	<b>83.22%</b>
6	Feature Combination - HED+Prewitt	82.70%	81.52%	84.46%	83.52%	<b>83.05%</b>
7	Feature Combination - Prewitt+Canny	80.37%	80.62%	82.17%	83.27%	<b>81.61%</b>
8	Feature Combination HED+Prewitt+Canny	-	81.14%	82.98%	84.24%	83.45%
9	RGB enhanced with Canny edges	71.11%	70.02%	79.53%	76.49%	74.29%
10	RGB enhanced with Prewitt edges	74.40%	76.73%	82.26%	79.99%	78.35%
11	RGB enhanced with HED edges	79.90%	83.39%	83.85%	80.84%	<b>82.00%</b>
12	RGB enhanced with Canny+HED edges	68.66%	76.98%	80.81%	74.35%	75.20%
13	RGB enhanced with HED+Prewitt edges	79.81%	79.39%	82.64%	82.75%	<b>81.15%</b>
14	RGB enhanced with Prewitt+Canny edges	77.34%	81.81%	82.14%	76.72%	79.50%
15	RGB enhanced with All edges combined	83.86%	83.72%	83.36%	84.44%	<b>83.85%</b>

Fig. 9. Comparison of F1 Score for 15 datasets across 4 classifiers

- 3) HED features performed the best as a single feature and in feature combinations.

#### IV. SUMMARY OF RESULTS

We trained and tested 4 models on 15 datasets with features only, features combination and RGB images enhanced with the same. The results show that the original image enhanced with combination of all three edges performed the best overall. Comparing the three edge detectors, performance on datasets with HED edges as single feature and its combination performed better. This could be attributed to the clear, thick edge features given by HED.

#### V. CONCLUSION AND FUTURE WORK

The project aimed at creating a robust, balanced augmented dataset from initial set of 27,000 ground truth data of textureless objects. Using that as baseline, we created 14 other datasets each of 340,000 images by using 3 edge detectors, Canny, Prewitt and HED. The initial seven were 3 single features and 4 of its combination dataset. The next seven were RGB images enhanced with the initial seven dataset. The experiments proved that the best performance was presented by the RGB images enhanced with all features combined. HED features proved to be performing the best among all features. Since the initial ground truth was with white background, the models performed really well on images with white background, whereas it could not perform when there was a

	Dataset	Training Accuracy	Validation Accuracy	Difference
1	Initial Augmented training data	64%	62%	2%
2	Features only using Canny	96%	78%	<b>18%</b>
3	Features only using Prewitt	89%	78%	<b>10%</b>
4	Features only using HED	88%	80%	8%
5	Feature Combination - Canny+HED	89%	80%	9%
6	Feature Combination - HED+Prewitt	87%	79%	7%
7	Feature Combination - Prewitt+Canny	89%	79%	<b>11%</b>
8	Feature Combination HED+Prewitt+Canny	-	80%	7%
9	RGB enhanced with Canny edges	68%	66%	2%
10	RGB enhanced with Prewitt edges	72%	70%	2%
11	RGB enhanced with HED edges	76%	73%	3%
12	RGB enhanced with Canny+HED edges	71%	69%	2%
13	RGB enhanced with HED+Prewitt edges	73%	71%	2%
14	RGB enhanced with Prewitt+Canny edges	68%	66%	2%
15	RGB enhanced with All edges combined	87%	79%	7%

Fig. 10. Comparison of Progressive Training and Validation accuracy to evaluate Overfitting for 15 datasets across 4 classifiers

	Dataset	SGD	Perceptron	Passive Aggressive I	Passive Aggressive II	Avg.
1	Initial Augmented training data	71%	69%	49%	71%	65%
2	Features only using Canny	83%	78%	84%	82%	<b>82%</b>
3	Features only using Prewitt	80%	84%	78%	90%	<b>83%</b>
4	Features only using HED	85%	77%	81%	84%	<b>82%</b>
5	Feature Combination - Canny+HED	84%	85%	84%	83%	<b>84%</b>
6	Feature Combination - HED+Prewitt	81%	84%	81%	84%	<b>82%</b>
7	Feature Combination - Prewitt+Canny	82%	84%	85%	86%	<b>84%</b>
8	Feature Combination - HED+Prewitt+Canny	78%	87%	85%	82%	<b>83%</b>
9	RGB enhanced with Canny edges	75%	69%	77%	75%	74%
10	RGB enhanced with Prewitt edges	67%	69%	77%	78%	73%
11	RGB enhanced with HED edges	78%	84%	81%	78%	<b>80%</b>
12	RGB enhanced with Canny+HED edges	69%	77%	86%	77%	77%
13	RGB enhanced with HED+Prewitt edges	75%	80%	79%	84%	79%
14	RGB enhanced with Prewitt+Canny edges	70%	83%	83%	78%	79%
15	RGB enhanced with All edges combined	87%	87%	86%	88%	<b>87%</b>

Fig. 11. Comparison of model performance on test set 1 with white background



Fig. 12. Sample images from test set 2

different background. In future work, we would like to perform the experiments that can generalize the comparison between features irrespective of background or clutter. This could be achieved by adding background and clutter to the training set or by removing background or clutter from the test set.

## VI. ACKNOWLEDGEMENT

The authors would like to thank our mentor, Gabriel Lugo Bustillo, Phd candidate, dept. of Computing Science, University of Alberta for his guidance and feedback throughout the research. We would also like to thank our advisors Dr. Irene Cheng and Dr. Anup Basu for their support and motivation to bring out the best in our research. We would also like to thank the researchers of previous work in the field of Textureless Object Recognition, which became the starting point for our research.

## VII. RESOURCES

Source Code: <https://tinyurl.com/v7bbkrk>

		<b>SGD</b>	<b>Perceptron</b>	<b>Passive Aggressive I</b>	<b>Passive Aggressive II</b>	<b>Avg.</b>
1	Initial Augmented training data	38%	38%	44%	31%	38%
2	Features only using Canny	6%	6%	19%	6%	9%
3	Features only using Prewitt	25%	25%	19%	25%	23%
4	Features only using HED	56%	31%	44%	44%	44%
5	Feature Combination - Canny+HED	31%	38%	50%	50%	42%
6	Feature Combination - HED+Prewitt	38%	44%	38%	44%	41%
7	Feature Combination - Prewitt+Canny	38%	19%	31%	19%	27%
8	Feature Combination - HED+Prewitt+Canny	38%	25%	31%	44%	34%
9	RGB enhanced with Canny edges	44%	38%	38%	38%	39%
10	RGB enhanced with Prewitt edges	31%	31%	31%	38%	33%
11	RGB enhanced with HED edges	44%	31%	38%	19%	33%
12	RGB enhanced with Canny+HED edges	31%	31%	44%	50%	39%
13	RGB enhanced with HED+Prewitt edges	44%	44%	50%	44%	45%
14	RGB enhanced with Prewitt+Canny edges	44%	44%	50%	38%	44%
15	RGB enhanced with All edges combined	56%	63%	25%	44%	47%

Fig. 13. Comparison of model performance on test set 2 with different background

## REFERENCES

- [1] Eggert, D. W., Bowyer, K. W., Dyer, C. R., Christensen, H. I. and Goldgof, D. B., "The Scale Space Aspect Graph," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, pp. 1114-1130 (1993). doi: 10.1109/34.244674.
- [2] Steger, C., "Occlusion, Clutter, and Illumination Invariant Object Recognition," International Archives of the Photogrammetry, Remote Sensing, and Spatial Information Sciences, Vol. 34, Part 3A, pp. 345-350 (2002).
- [3] Cyr, C. M. and Kimia, B. B., "A Similarity-based Aspect-graph Approach to 3D Object Recognition," International Journal of Computer Vision, Vol. 57, No. 1, pp. 522 (2004).
- [4] Maini, Raman, and Himanshu Aggarwal. "A comprehensive review of image enhancement techniques." arXiv preprint arXiv:1003.4053 (2010).
- [5] Al-Amri, Salem Saleh, N. V. Kalyankar, and S. D. Khamitkar. "Image segmentation by using edge detection." International journal on computer science and engineering 2, no. 3 (2010): 804-807.
- [6] Hinterstoisser, Stefan, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. "Multimodal templates for real-time detection of textureless objects in heavily cluttered scenes." In 2011 international conference on computer vision, pp. 858-865. IEEE, 2011.
- [7] Hinterstoisser, Stefan, Cedric Cagniart, Slobodan Ilic, Peter Sturm, Nassir Navab, Pascal Fua, and Vincent Lepetit. "Gradient response maps for real-time detection of textureless objects." IEEE Transactions on Pattern Analysis and Machine Intelligence 34, no. 5 (2011): 876-888.
- [8] Ulrich, M., Wiedemann, C. and Steger, C., "Combining Scale-space and Similarity-based Aspect Graphs for Fast 3D Object Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 34, No. 10, pp. 1902-1914 (2012). doi: 10.1109/TPAMI.2011.266.
- [9] Tombari, Federico, Alessandro Franchi, and Luigi Di Stefano. "BOLD features to detect textureless objects." In Proceedings of the IEEE International Conference on Computer Vision, pp. 1265-1272. 2013.
- [10] Fischer, Jan, Richard Bormann, Georg Arbeiter, and Alexander Verl. "A feature descriptor for textureless object representation using 2D and 3D cues from RGB-D data." In 2013 IEEE International Conference on Robotics and Automation, pp. 2112-2117. IEEE, 2013.
- [11] Bali, Akanksha, and Shailendra Narayan Singh. "A review on the strategies and techniques of image segmentation." In 2015 Fifth International Conference on Advanced Computing & Communication Technologies, pp. 113-120. IEEE, 2015.
- [12] Janani, P., J. Premalatha, and K. S. Ravichandran. "Image enhancement techniques: A study." Indian Journal of Science and Technology 8, no. 22 (2015).
- [13] Hodan, Tomáš, Xenophon Zabulis, Manolis Lourakis, Štěpán Obdržálek, and Jiří Matas. "Detection and fine 3D pose estimation of textureless objects in RGB-D images." In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4421-4428. IEEE, 2015.
- [14] Makandar, Aziz, and Bhagirathi Halali. "Image enhancement techniques using highpass and lowpass filters." International Journal of Computer Applications 109, no. 14 (2015): 12-15. 15
- [15] Wang, Wei, Lili Chen, Ziyuan Liu, Kolja Kühnlenz, and Darius Burschka. "Textured/textureless object recognition and pose estimation using RGB-D image." Journal of Real-Time Image Processing 10, no. 4 (2015): 667-682.
- [16] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster r-cnn: Towards real-time object detection with region proposal networks." In Advances in neural information processing systems, pp. 91-99. 2015.
- [17] Jiang, Lixing, Artur Koch, and Andreas Zell. "Object recognition and tracking for indoor robots using an RGB-D sensor." In Intelligent Autonomous Systems 13, pp. 859-871. Springer, Cham, 2016.
- [18] Chan, Jacob, Jimmy Addison Lee, and Qian Kemao. "Border: An oriented rectangles approach to textureless object recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2855-2863. 2016.
- [19] Wang, Yida, and Weihong Deng. "Self-restraint object recognition by model based CNN learning." In 2016 IEEE International Conference on Image Processing (ICIP), pp. 654-658. IEEE, 2016.
- [20] Verma, Nishchal K., Teena Sharma, Shreedharkumar D. Rajurkar, Rakesh Ranjan, and Al Salour. "Vision based counting of textureless objects using shape and color features." In 2016 11th International Conference on Industrial and Information Systems (ICIIS), pp. 253-258. IEEE, 2016.
- [21] Peng, Yixin, Lili Bao, and Ling Pi. "Object (s)-of-interest segmentation for images with inhomogeneous intensities based on curve evolution." Neurocomputing 195 (2016): 13-18.
- [22] Thoduka, Santosh, Stepan Pazekha, Alexander Moriarty, and Gerhard K. Kraetzschmar. "RGB-D-Based Features for Recognition of Textureless Objects." In Robot World Cup, pp. 294-305. Springer, Cham, 2016.
- [23] Chan, Jacob, Jimmy Addison Lee, and Qian Kemao. "BIND: binary integrated net descriptors for textureless object recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2068-2076. 2017.
- [24] Hodan, Tomáš, Pavel Haluza, Štěpán Obdržálek, Jiri Matas, Manolis Lourakis, and Xenophon Zabulis. "T-LESS: An RGB-D dataset for 6D pose estimation of textureless objects." In 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 880-888. IEEE, 2017.
- [25] Bloice, Marcus D., Christof Stocker, and Andreas Holzinger. "Augmentor: an image augmentation library for machine learning." arXiv preprint arXiv:1708.04680 (2017).
- [26] Xiang, Yu, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes." arXiv preprint arXiv:1711.00199 (2017).
- [27] Jeong, Hyek-June, Kyeong-Sik Park, and Young-Guk Ha. "Image preprocessing for efficient training of YOLO deep learning networks." In

- 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 635-637. IEEE, 2018.
- [28] Tsai, Chi-Yi, and Chao-Chun Yu. "Real-time textureless object detection and recognition based on an edge-based hierarchical template matching algorithm." *J. Appl. Sci. Eng.* 21, no. 2 (2018): 229-240.
- [29] Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).
- [30] Huang, Rui, Jinan Gu, Xiaohong Sun, Yongtao Hou, and Saad Uddin. "A Rapid Recognition Method for Electronic Components Based on the Improved YOLO-V3 Network." *Electronics* 8, no. 8 (2019): 825.
- [31] Agustsson, Eirikur, Jasper RR Uijlings, and Vittorio Ferrari. "Interactive Full Image Segmentation by Considering All Regions Jointly." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 11622-11631. 2019.
- [32] Fang, Wei, Yewen Ding, Feihong Zhang, and Victor S. Sheng. "DOG: A new background removal for object recognition from images." *Neurocomputing* 361 (2019): 85-91.
- [33] Li, Yixing, and Fengbo Ren. "Light-Weight RetinaNet for Object Detection." arXiv preprint arXiv:1905.10011 (2019).
- [34] Park, Kiru, Timothy Patten, Johann Prankl, and Markus Vincze. "Multi-Task Template Matching for Object Detection, Segmentation and Pose Estimation Using Depth Images." In 2019 International Conference on Robotics and Automation (ICRA), pp. 7207-7213. IEEE, 2019.
- [35] He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. "Mask r-cnn." In Proceedings of the IEEE international conference on computer vision, pp. 2961-2969. 2017.
- [36] Yu, Hancheng, Haibao Qin, and Maoteng Peng. "A Fast Approach to textureless Object Detection Based on Orientation Compressing Map and Discriminative Regional Weight." *Algorithms* 11, no. 12 (2018): 201.
- [37] Xie, Saining, and Zhuowen Tu. "Holistically-nested edge detection." In Proceedings of the IEEE international conference on computer vision, pp. 1395-1403. 2015.



**Dhara Pancholi** is currently a MSc. in Computing Science specializing in Multimedia at University of Alberta. She completed her bachelor's degree in Computer Science from India. Apart from different subjects that she learned, her interest lies in computer graphics and animation. She has also worked on different IOT projects in her bachelors and currently pursuing 3 projects at university of Alberta as course projects.

## IX. APPENDIX

### VIII. AUTHOR BIOGRAPHY



**Frincy Clement** is an Artificial Intelligence enthusiast who is currently pursuing her MSc. in Computing Science at the University of Alberta. She received her BTech in Computer Science and Master in Business Administration from top universities in India. She has 6+years of work experience and is a seasoned software developer with experience working in several application as well as research-based projects in AI.



**Kirtan Shah** is a student pursuing Masters in the field of Computing Science at University of Alberta. He completed his Bachelor's in the field of Computer Engineering in India. He has worked on projects that integrate IOT application for the real-world needs during his bachelor's. His interest is to apply various techniques to solve real world problems. He likes to play chess as well.

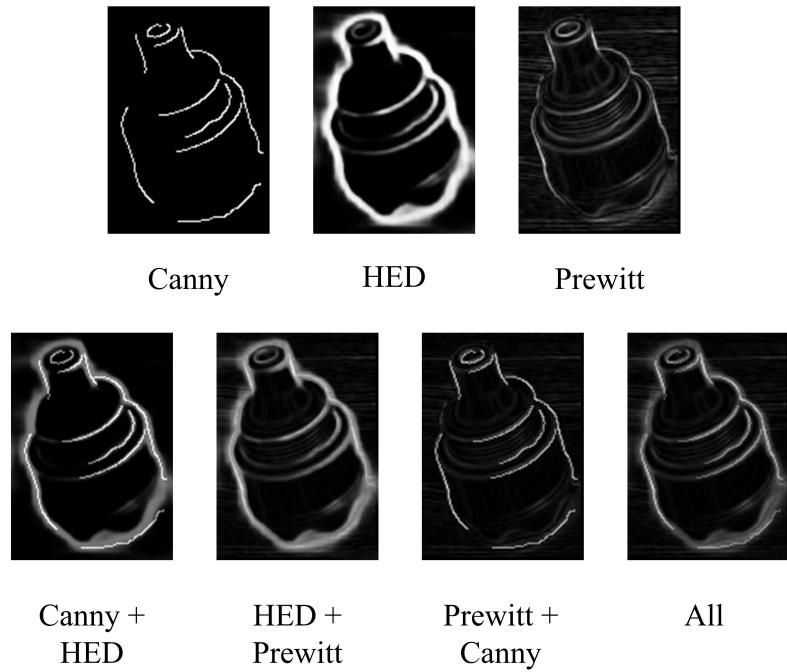


Fig. 14. Sample of test set 2 after initial preprocessing for features/feature combinations

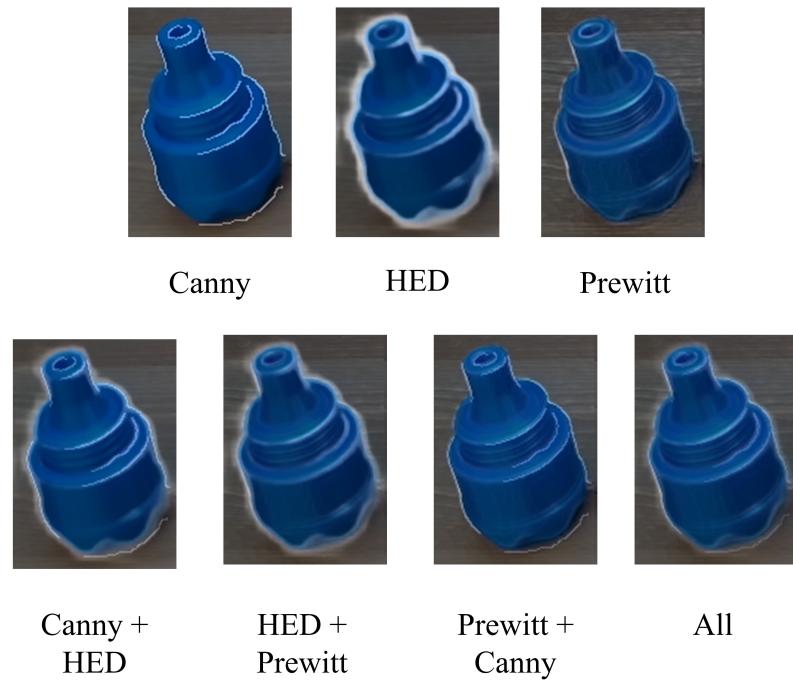


Fig. 15. Sample of test set 2 after initial preprocessing for RGB enhanced with features/feature combinations