Internship at Shibata Lab

Udit Jain

Computer Science and Engineering,
IIT-Delhi

July 13, 2018

Grasping Cloth by Baxter Arm

Subtasks

1. Detecting Segmented mask.
2. Bounding Box from Kinect Image .
3. Generating pickup coordinates 3D from mask and PCD data.
4. Navigating to point and gripping cloth.
5. Presenting all controls in a User Interface.

Task 1

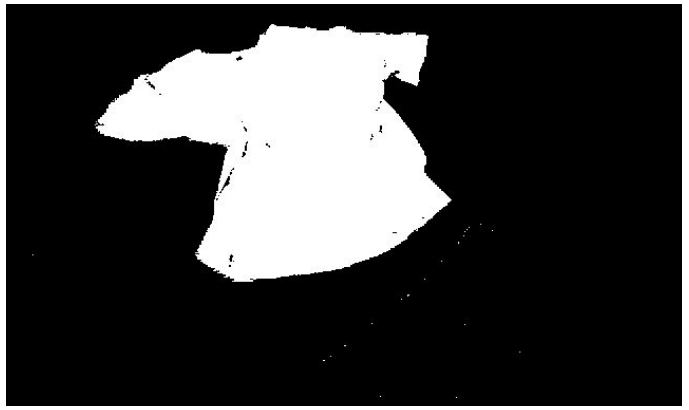
Segmenting Cloth Images

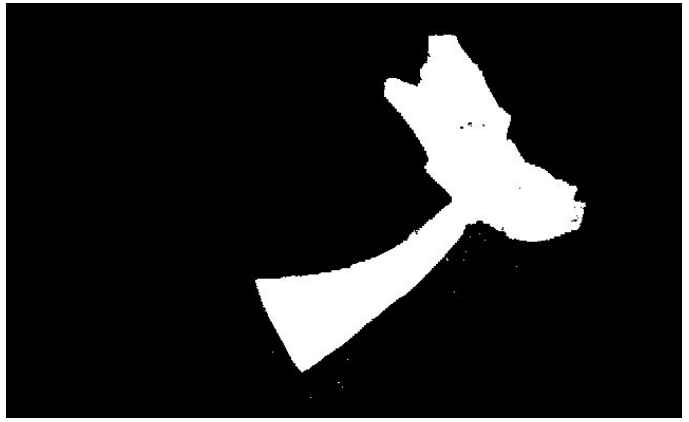
Aim : Generating a mask of the cloth region.

Segmentation

Input

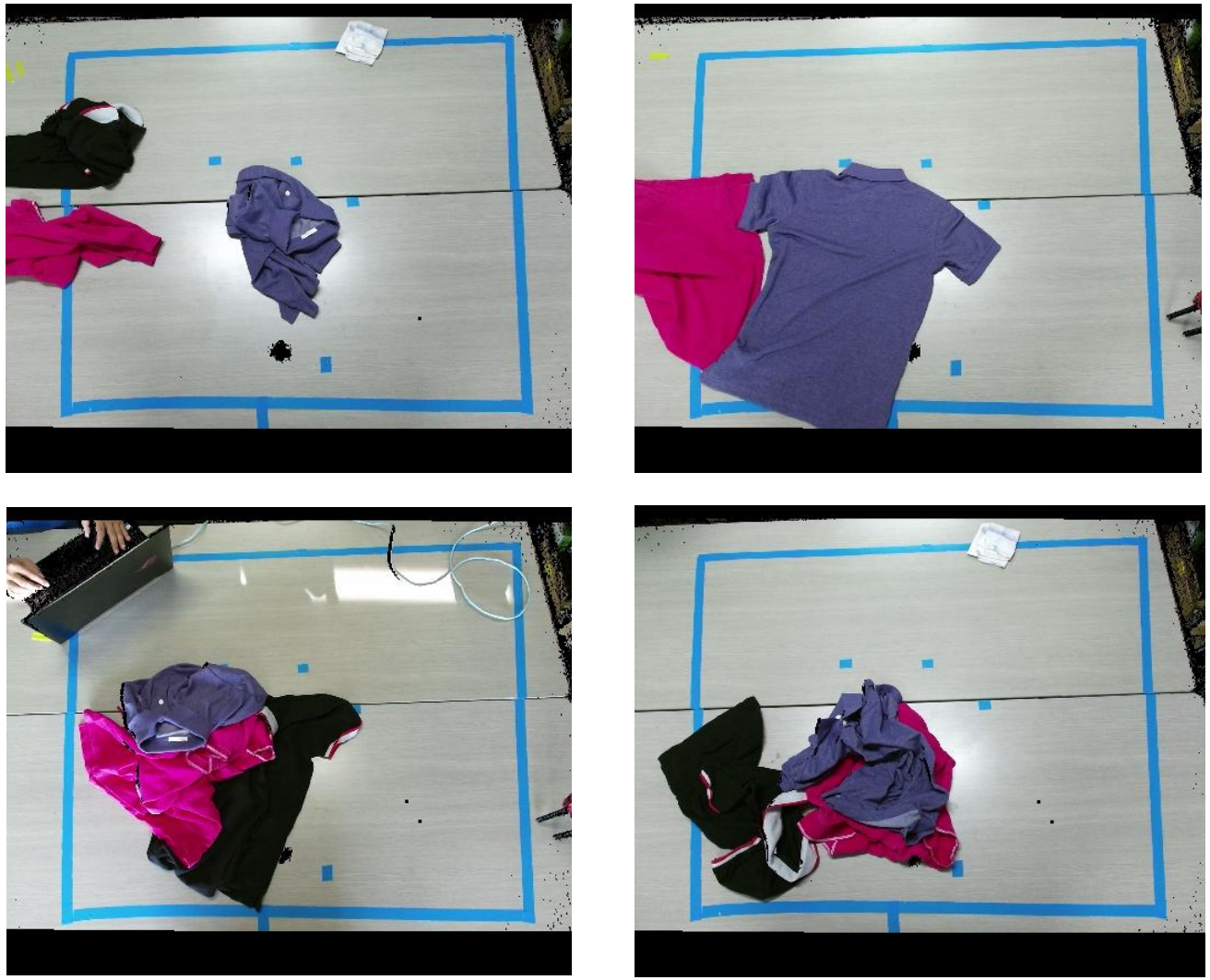
Model Outputs





Sample Images from Kinect

- **Noisy** images and PCD.
- **Unlabeled**, challenge to using deep network.
- Final metric for model's performance **evaluation**.



Dataset

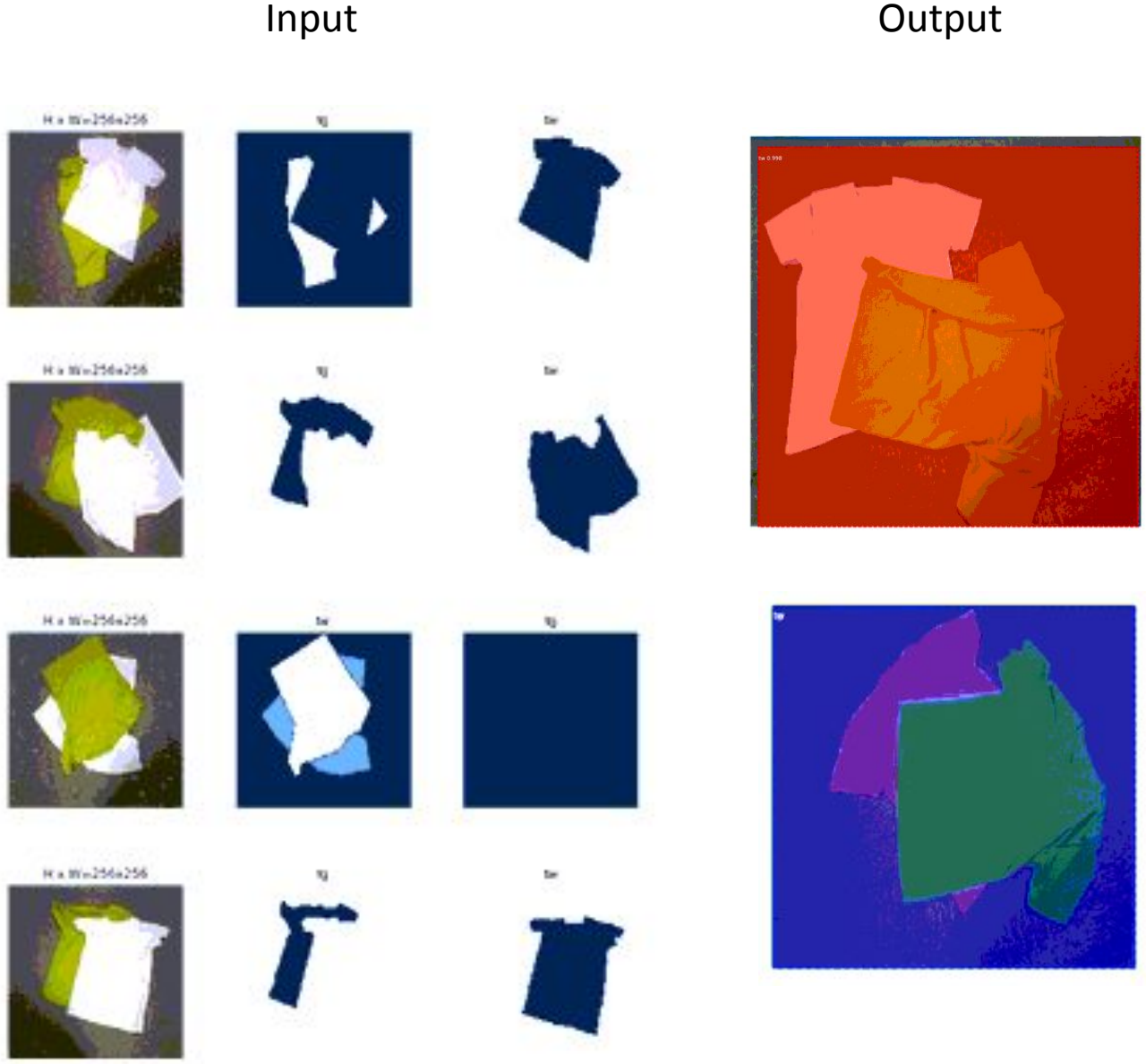
- Total 350 images :
  1. 300 - 2 T-shirts
  2. 50 - 3 T-shirts
- Only **100 labelled** images.
- Negligible **Noise**.
- **JSON** labels, **hard to parse** and feed into deep networks.
- Best labels for segmentation are 1 0 masks.




Previous Model :

Mask RCNN

- Very **bad** Mean Average Precision score.
- Unsuitable for grasping application.
- How to improve it ?



Approach 1 : Using Deep Networks

SEGNET, Deep-Lab

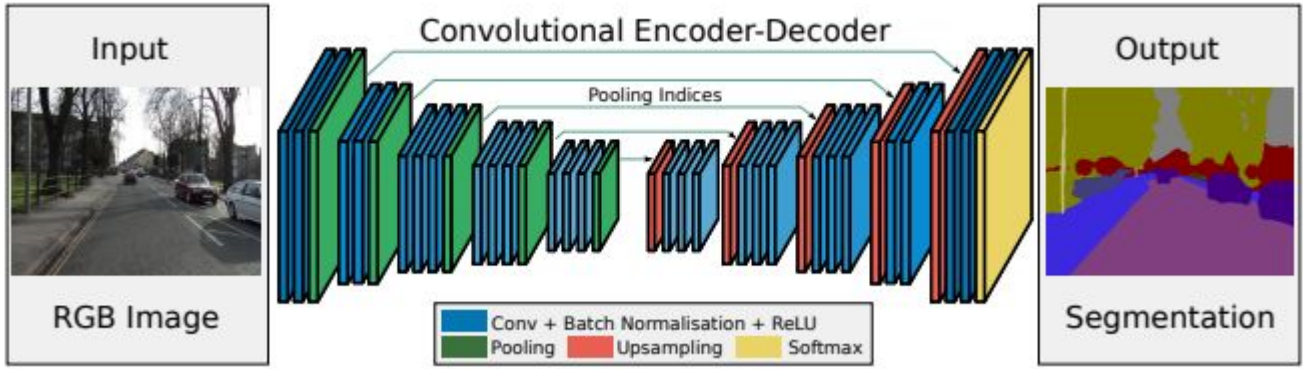
- Pros
  1. Best for **rigid** objects.
  2. Can give good results with relatively less data.
- Cons
  1. **Un-adaptive** to clothes (deformable).

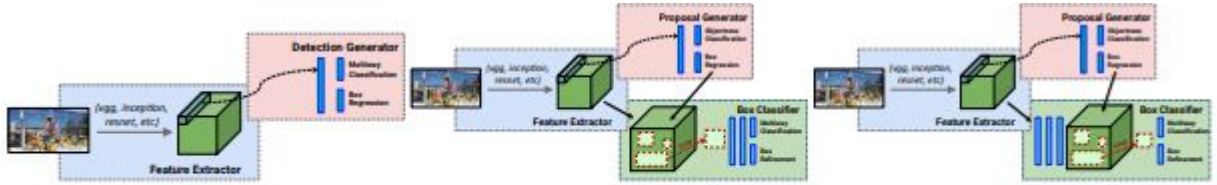
Mask - RCNN

- Pros
  1. Adapts to **deformable** objects.
  2. **Instance segmentation**.
- Cons
  1. **More** training data to **converge**.

SEGNET

- Poor results due to flexible and deformable nature of clothes.
- **Bad** instance segmentation.
- **Unsatisfactory** results on Kinect data.

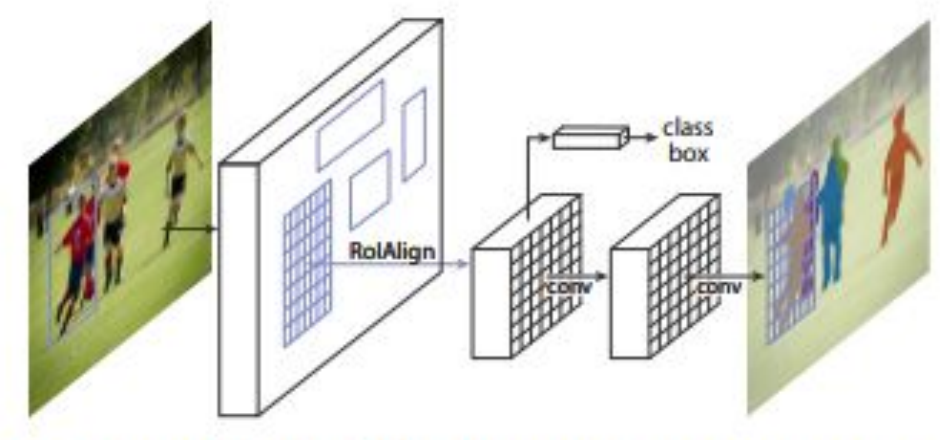




[SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation]

Mask RCNN

- Trained on **augmented labelled data**, with pre-trained **coco weights**. For 250 epochs.
- Batch Norm to generalize to Kinect data.
- Testing MAP: ~0.43.

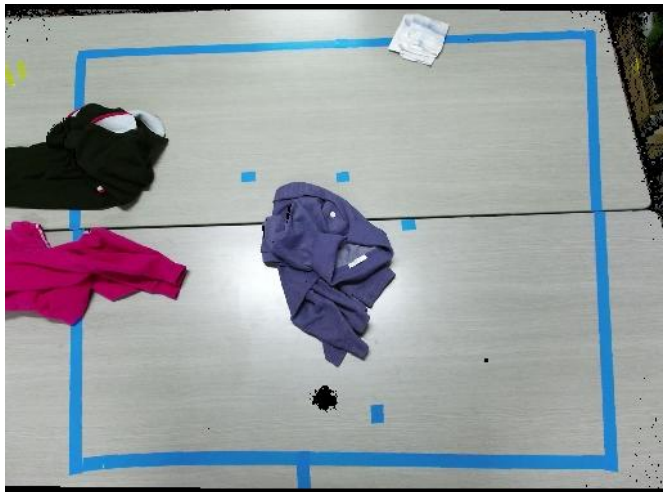
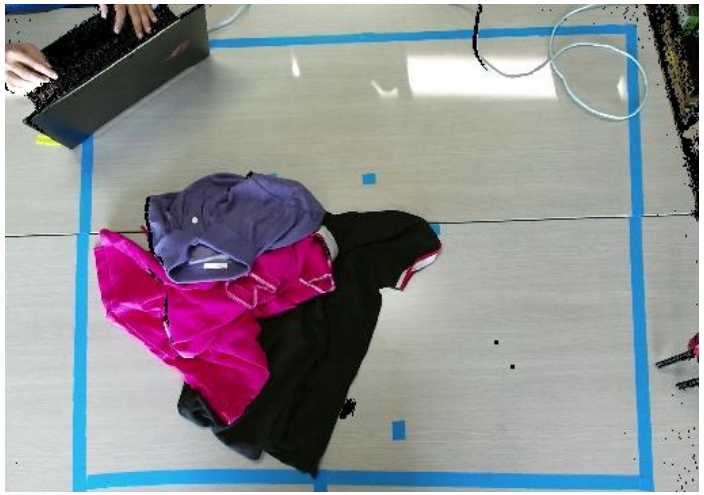
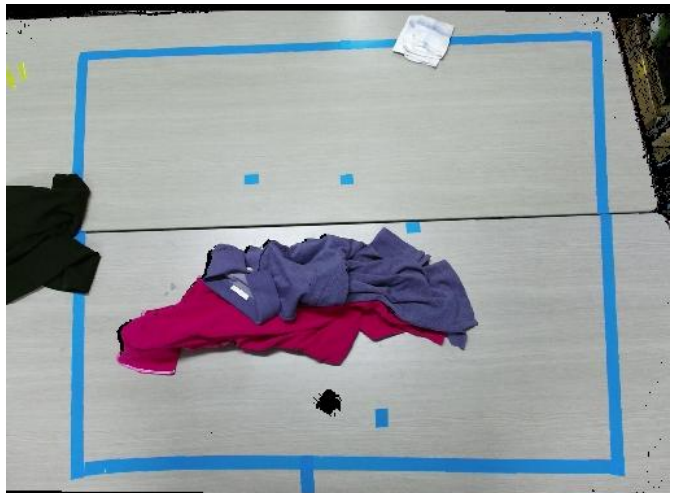
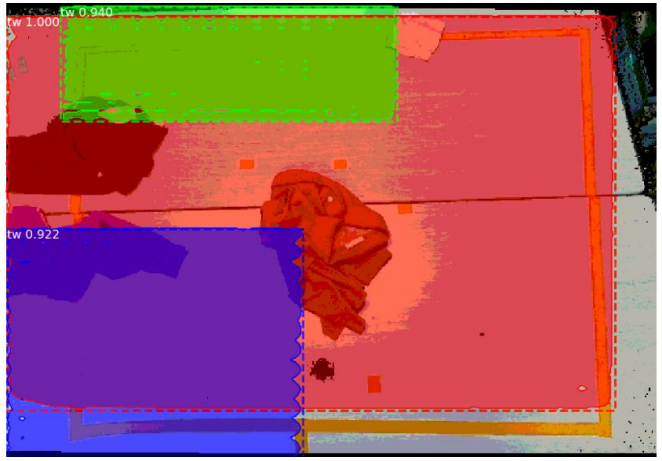


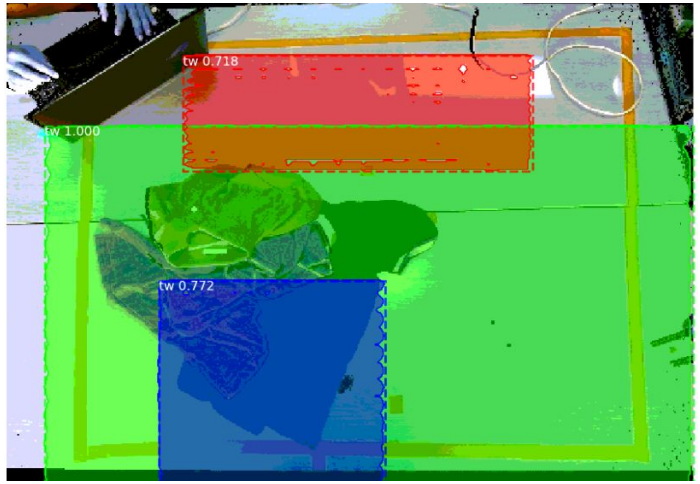
Sample Output on COCO

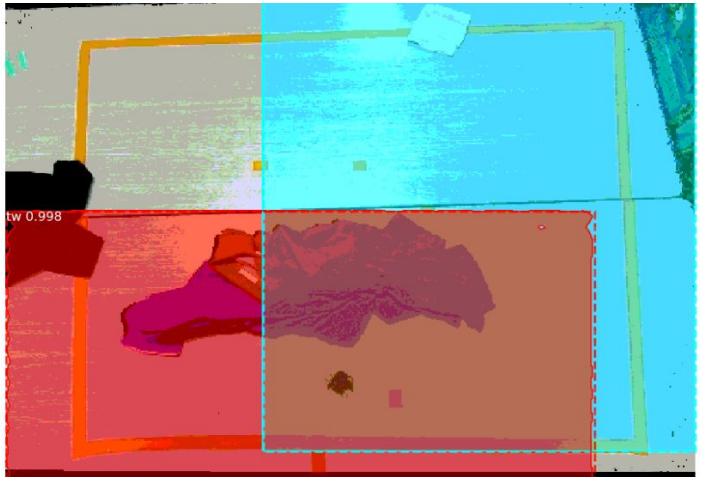
[Mask R-CNN]

Results from improved model :

Better but still unsatisfactory

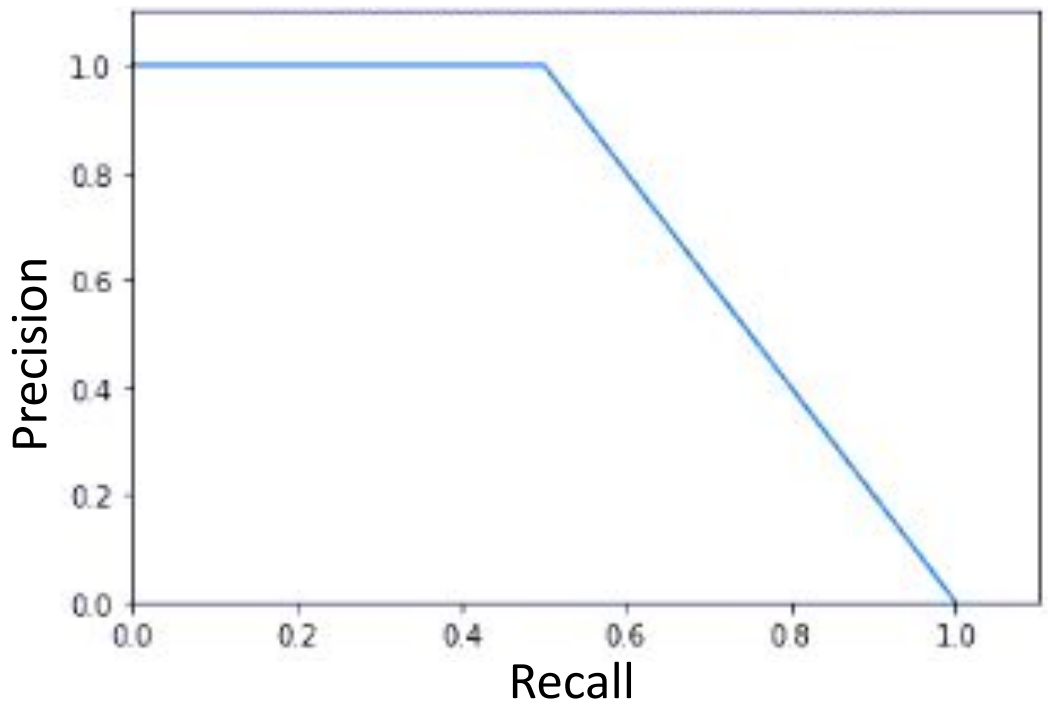




Performance Analysis Mask RCNN

- **Bad** mean average precision.
- Doesn’t converge, ~6M parameters. Less Training data.

```
mAP:  0.43250000178813935
```



Challenges

- **Less Training Data** :100 original labelled images. Model pre-trained on ~2.5M labelled object instances.
- **Convergence** : Complex model, large number parameters.

Solution:

I label the 350 image dataset and 2000 image dataset [Kinect].
[Approach 2 discussed later]

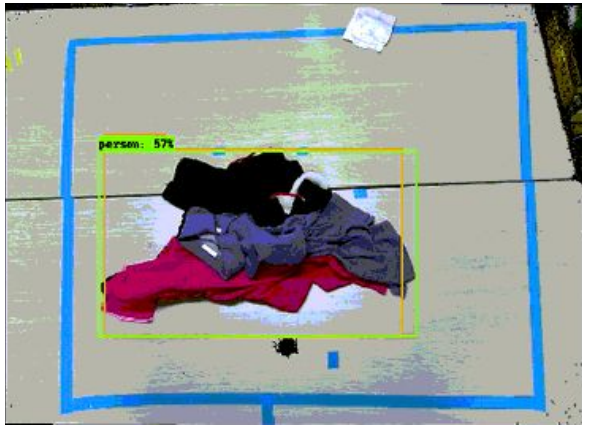
Task 2

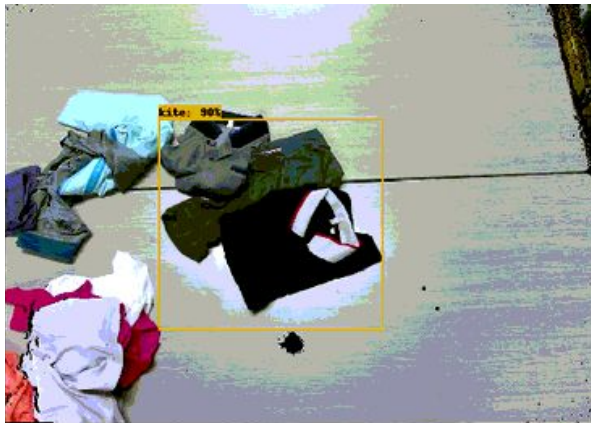
Bounding Box Detection in Cloth Images

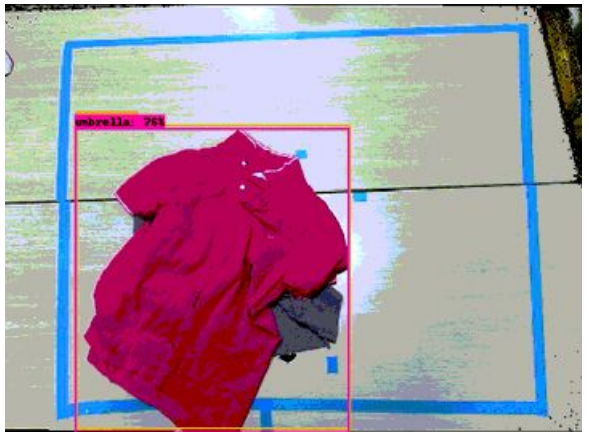
Aim : Generating cloth covering bounding box.

Object Detection Network

- Pros
  1. **Full object coverage:** More region to check for z values.
  2. Better coupled with good Point picking algorithm.
- Cons
  1. **Doesn't** separate instances.
  2. **Blank space**: Picking error prone. Baxter misses cloth.







Task 3

Finding Pickup coordinates

Aim : 3D coordinates from 2D Kinect image and PCD.

Task

Input : Extremity ROI BB mapped on 2D image and Kinect PCD.

Output : < x, y, z > for grasping.

Challenges

- **Noise** in PCD. Averaging multiple PCD snapshots doesn't help much, because **Systematic** error.
- **Outliers** : Problem with the Max-Z approach.
- **Corrected** : Percentile Method.

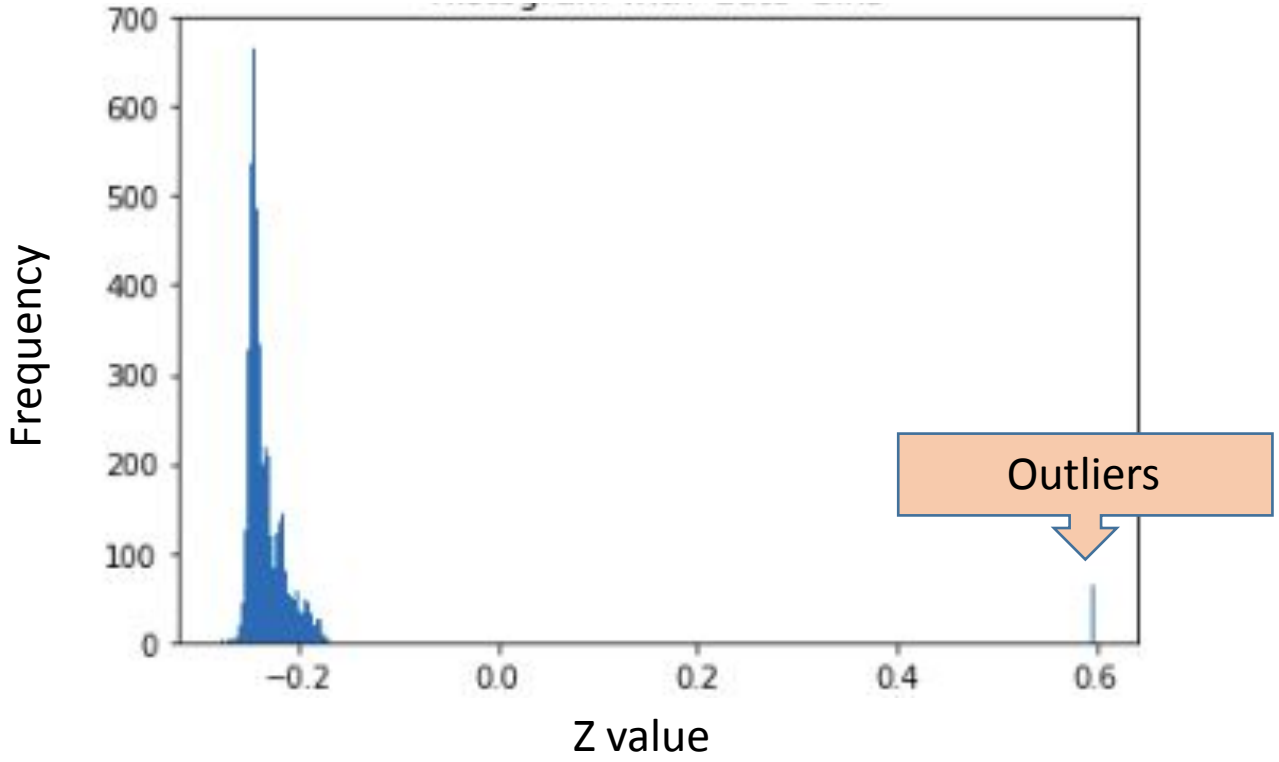
Distribution of Z - coordinates

Outliers at end are noise due to **reflection** and other **lighting conditions**.

Focus on first plateau's max

Another Possible Approach:

Color tracking [of Gripper] to improve algorithm in future. From Kinect/ Baxter Arm camera.



Histogram of Z coordinates

Task 4

User Interface

Aim : Making a UI for easy demonstration.

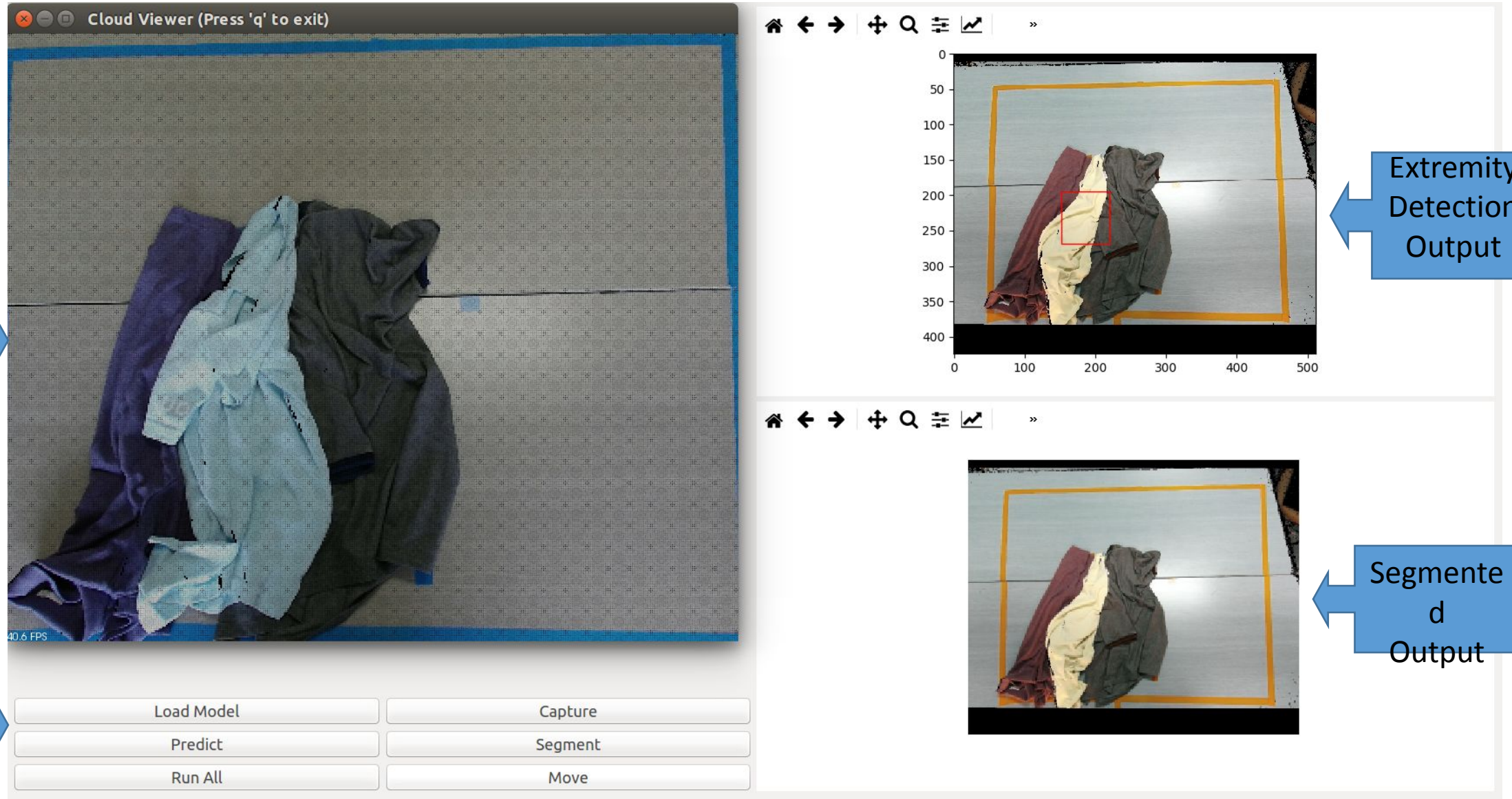
Task

- **Coordinating :** ROI detection, segmentation and movement of Robot.
- **Interface :** All at one place.
- **Generalized enough :** to other models.

Challenges

- Kinect output stream has callbacks and event listeners in C++, **difficult** to **migrate** to Python based software.
- Both Deep learning and ROS libraries in the same program, **conflicting dependencies**.

Made a generalized UI , which can incorporate any model's output, not just segmentation and ROI detection.



Task 5

Labelling images for Segmentation

Aim : Generate more training data.

Task

- **Not enough labeled data.**
- Algorithm : Generating mask of 350 + 2000 images.

Challenges

- No generalization to segment other datasets.

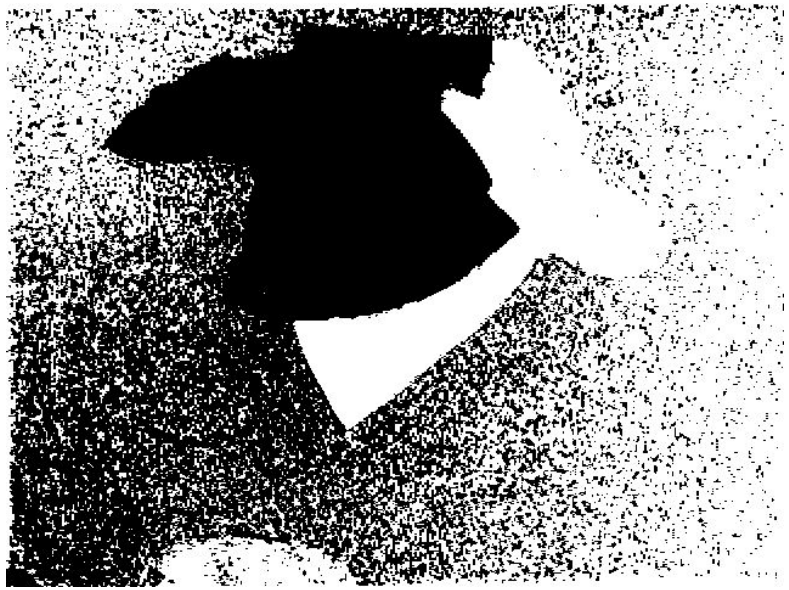
Algorithms tested

1. RGB color space segmentation with K-means clustering
   - Patchy and noisy masks.
   - Uniformity introduced after K-means but still sub-optimal.

2. Otsu's Algorithm
   - Better performance on 350 [cleaner] image dataset.
   - Bad on Kinect images.

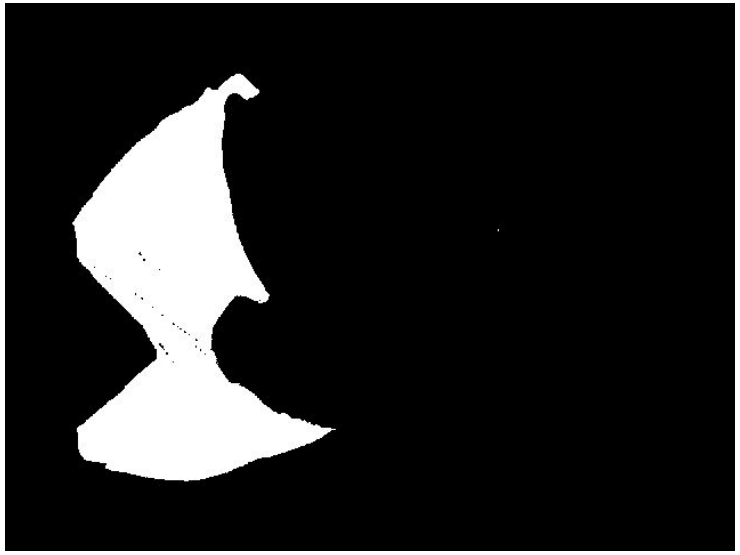
3. L a b color space

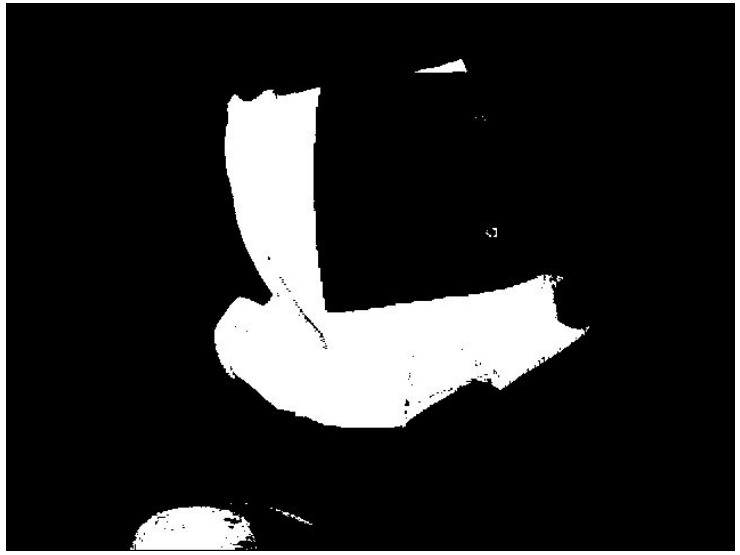
- Noisy masks.
- **Best** for Bi – Tri modal distribution in L a b space.

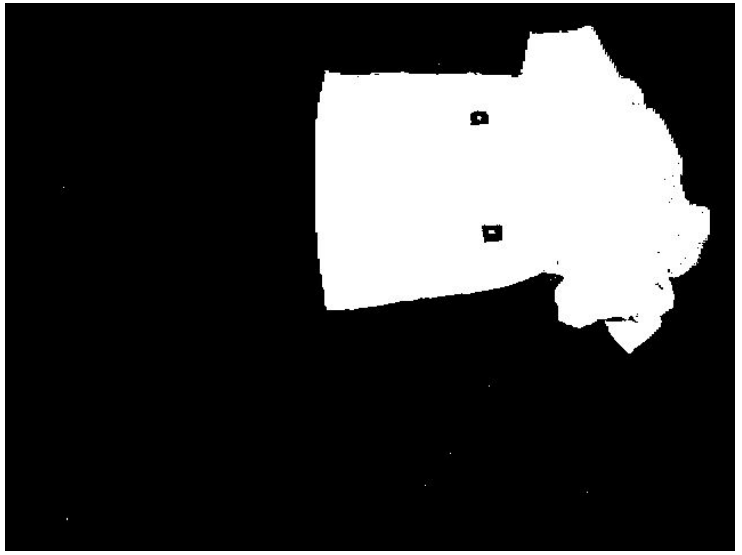


4. H S V color space

- Thresholded in HSV space, **cleanest masks**.
- **Least generalizable**.
- Problem if more T-shirts.



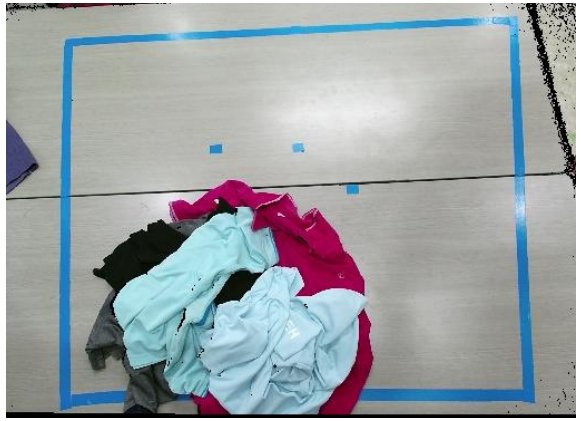


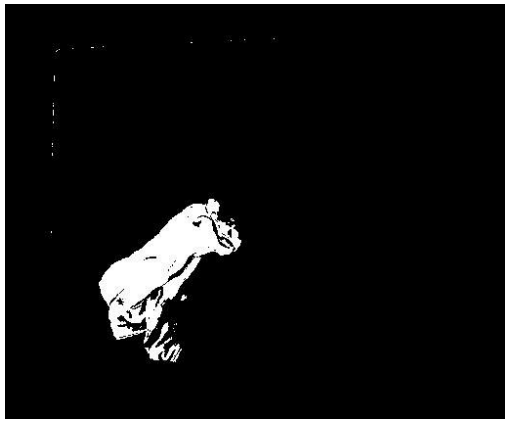


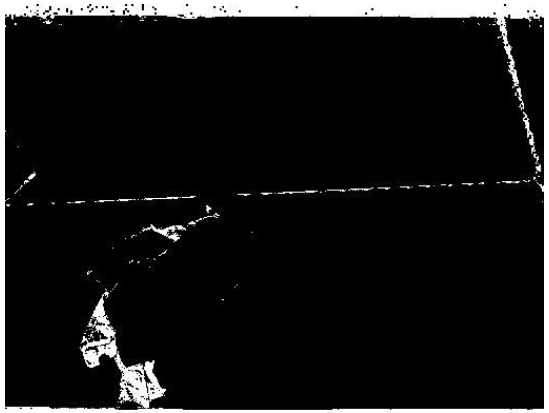
Test on Kinect images

Labels:

1. Black 2. Dark Turquoise Blue 3. Dark Grey
4. Dark Pink 5. Light Turquoise Blue 6. Purple







Future Work

- Augment images and labels and train Deep Model.
- Generalized and better segmentation and ultimately, better **picking**.

Thank You Everyone!