OscarNet: Detecting and Extracting Garbage

Anirudh Rathore
Department of Computer Science
University of Colorado Boulder
anirudh.rathore@colorado.edu

Aser Garcia
Department of Computer Science
University of Colorado Boulder
aser.garcia@colorado.edu

Himanshu Gupta
Department of Computer Science
University of Colorado Boulder
himanshu.gupta@colorado.edu

James Hunter Hobbs
Department of Computer Science
University of Colorado Boulder
james.hobbs@colorado.edu

Abstract

The world is currently in the grips of a trash crisis. It is estimated that the US produces approximately 234lb (106.2kg) of plastic waste per person annually(1). The enormous amount of waste produced does not get discarded properly and ends up on the roadside, or in parks and wildlife areas. Our goal is to address this issue by developing an autonomous system that classifies and then removes trash from public spaces. We implemented a Mask R-CNN based model to detect garbage in images and create segments over it. We use the transfer learning approach to reduce training time by using a model trained on the famous COCO dataset(2). In conjunction, we use a pre-existing image depth estimator to aid autonomous navigation(3). Our model OscarNet, named after the garbage loving character from the show Sesame Street, was trained on the dataset that we curated.

1 Introduction

Our brain instantly recognizes the objects in an image. However, it seems like a really daunting task to teach a machine to identify these objects and their shapes. With the recent advances in computational power and efficient deep learning algorithms, this computer vision problem has become a whole lot easier to solve. Object detection technology has seen a rapid adoption rate in various and diverse industries. It helps self-driving cars safely navigate through traffic, spots violent behavior in a crowded place, assists sports teams in analyzing and building scouting reports, ensures proper quality control of parts in manufacturing, among many, many other things.

Deep learning has so much potential in the object detection space. We can pass the original image through a deep neural network and reinforce a deep learning algorithm to give predictions as close to the original bounding box in the training data as possible. Leveraging this process will ensure that the algorithm classifies tighter and more precise bounding box predictions in comparison to earlier naive approaches. People have applied Convolutional Neural Networks (CNNs) and their variants on the GINI dataset and obtained amazing results for garbage detection. CNN's have become the go-to method for solving any image data challenge, and their use is being extended to video analysis, as well. Any data that has spatial relationships are ripe for applying CNN.

In CNN, an image is provided to the input and sent through various convolutions and pooling layers. The neural network (NN) infers an object classification and can extend to detect multiple objects in an image. We divide the image into various regions and consider each sub-region as a separate image, pass all these sub-regions(images) to CNN and classify them into their respective classes. After detecting objects in the sub-regions, we can combine all of them to get back the original image with the detected objects. The problem with this approach is that the objects in the image generally have different aspect ratios, shapes, and spatial locations, and this makes it difficult for the algorithm to learn tight bounding boxes. This broad-bounding box problem can be mitigated by dividing the image in some number of regions, but this increases computational time significantly. To avoid this and reduce the number of regions, we use region-based CNN or RCNN, which selects the regions using a proposed method. Further improvements were performed on RCNN to make it even faster. Fast RCNN and Faster RCNN are deep neural network algorithms developed to improve the performance and speed of RCNN.

For a self-driving car, the computer vision technique used for simple object detection is too limited because it merely detects an object and draws a box-shaped figure around it. If there's a sharp turn in the road ahead, the model will detect it and draw a rectangular box around it. The model responsible for driving the car could have trouble understanding the shape of the road would be unable to figure out if it should turn or go straight. To avoid this, we need a technique that can detect the exact shape and turn of the road so our self-driving car system can safely navigate the sharp corners. Image segmentation creates a pixel-wise mask for each object in the image. This technique gives us a far more granular understanding of the object(s) in the image. Mask RCNN is the latest state-of-the-art architecture that is used to build such a system. The Mask R-CNN framework extends Faster R-CNN. For any given image, Mask R-CNN, in addition to the class label and bounding box coordinates for each object, will also return the object mask.

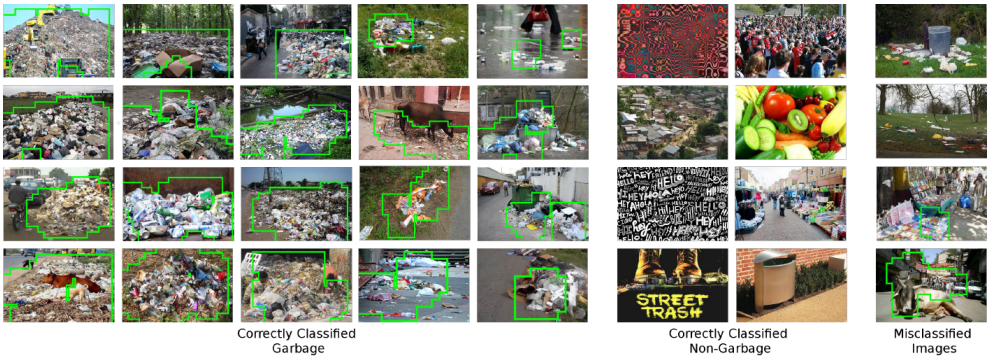

Figure 1: Performance of SpotGarbage on GINI Dataset
(4)

Estimating the depth of objects from images is another critical problem in the field of computer vision. A solution to this problem has its offerings towards other challenging problems like recognition, pose estimation, 3D modeling, autonomous driving, foreground detection, selective blur, and semantic labeling. While estimating depth from a scene image is straightforward to human eyes, it poses challenges to an automated computer vision system. Recent advances in deep neural networks and CNN architecture has allowed many researchers to address this problem with CNN.

Putting these two methods together can help in making a system understand object orientation and location and can then be used to develop systems that can perform simple tasks autonomously.

2 Related Work

- SpotGarbage, an android app developed by Mittal et al. (4), uses deep learning to detect if there is garbage present in an image. Due to the minimal resources available for computing and memory in smartphones, the model required a low memory footprint and quick image classification. The seemingly random characteristics of garbage, such as varied color, texture, and shape, were used to differentiate from non-garbage objects. In essence, the model predicts if there is garbage on individual patches in the image and then combines those patches to form an overall mask on the region. After training several models, the one that out-performed featured a fully connected network without local response normalization. This model, deemed Garbage Network (GarbNet), produced an accuracy of 87.69% and a prediction time of 1.50 seconds.
- In 2018, Ying Wang et al. proposed a different model based on Residual Neural Networks (ResNet) (5). In their paper, they highlighted the fact that as cities get more advanced, pollution control will need to emphasis on a higher level. A model for garbage detection was developed using ResNet, which avoids vanishing gradients by skipping over layers in the network. They used pre-trained model parameters along with training the model on the GINI dataset and other typical urban scenes. The model showed improvement in the detection of the garbage when objects with similar texture and color were present, such as the shirt of a person or a small vehicle. The model achieved 6-second detection with 89% average accuracy on the GTX750ti GPU. The model was able to create regions where the garbage was in the image. Reducing the time of the prediction is still an open challenge.
- Liu et al. (6) propose an automatic garbage detection system based on deep learning and narrow-band Internet of things. The system automatically detects and identifies decoration garbage directly in the front-end embedded monitoring module, and manages thousands of monitoring front-ends through the narrow-band Internet of Things (IoT) and background server. In the front-end embedded module of the system, the improved YOLOv2 network model performs garbage detection and recognition. The average target box dimension clustering and classification network pre-training lead to improvements in the YOLOv2 model performance. The system remains lightweight by replacing the feature extraction network and other methods. The lightweight YOLOv2 network is optimized and ported to an embedded module. The experiment test shows that, compared with the traditional monitoring system, the cost reduces by more than half, which can effectively save resources, and the accuracy of detection and recognition improves.
- Song et al. (7) used depth along with RGB image channels to detect specific scenes from images. An RGB-Depth CNN model was developed in contrast to commonly used models that separate into RBG-CNN and Depth-CNN to then classify an image. The model increased the accuracy from 38.9% on a state-of-the-art model to 42.4% with the RBG-Depth fusion model. The final model implemented support vector machines (SVMs), leading to a fully connected RGB-Depth-CNN with linear SVMs. The proposed model beats the state of the art model by 1.9%, where the state-of-the-art model is a fine-tuned Places-CNN with Recurrent Convolutional-NNs. We hope to utilize a similar strategy for our classifier with depth estimation on the monocular GINI dataset.

3 Dataset

In this project, we used images from the GINI dataset and from downloaded commercial licensed images from Flickr. We mainly focused on images that contained garbage bags and plastic waste.

3.1 GINI Dataset

The Garbage in Images (GINI) Dataset was first introduced by Mittal et al. (4) as part of their SpotGarbage paper. The dataset is a collection of several in-the-wild images containing garbage. Each image includes annotations with perceived levels of severity and biodegradability. It contains a compilation of 2561 images.

The content of this dataset are

- Images crawled using garbage related queries.
- Images crawled using non-garbage related queries.
- Garbage queried images ambiguously annotated by users.

Although this dataset is large, popular and available, it only has bounding box annotations. We need outline/segment annotations around garbage in images. For this reason, we created our own dataset and manually annotated images to draw outlines around garbage in every image.

Figure 2: Examples of images in the GINI dataset

3.2 Creating our Own Dataset

We created our own dataset, SS Garbage or Sesame Street garbage. We shortlisted images from the GINI dataset by giving priority to images with garbage bags and plastic waste. We downloaded similar images from Flickr and generated a set of 184 images. To achieve our goal of training a model, we needed to create segmented areas of the images for every individual group of garbage, specifically garbage bags, in the image.

3.3 Annotations

To draw outlines around garbage in images, we used VGG Image Annotator developed by Dutta et al.(8) VGG Image Annotator allows us to place dots (shown in red) over an image, and it will automatically connect the dots with lines, eventually encapsulating the region we would like to mask.

Figure 3: Examples of images in SS Garbage dataset

We do this for every instance of garbage in an image. We got a json file that had information on segments in each image.

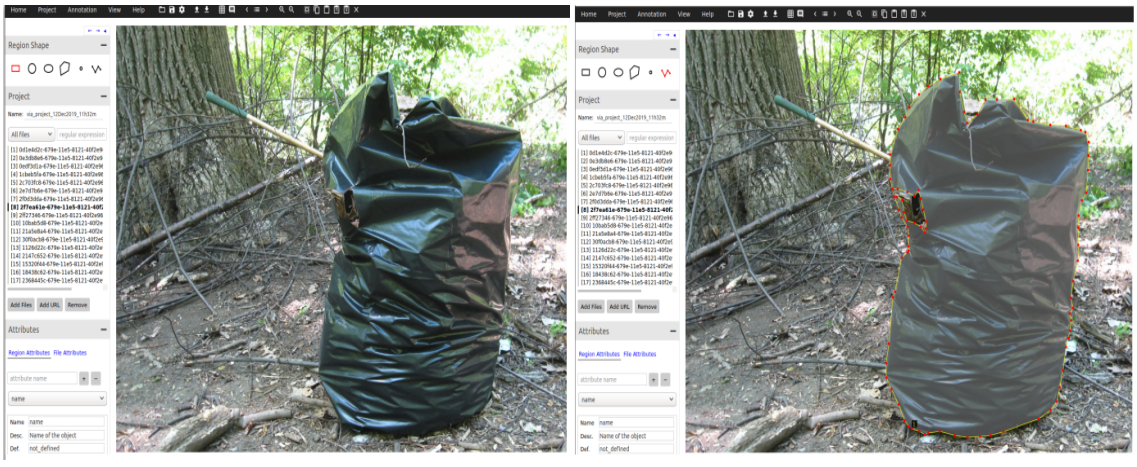

Figure 4: VGG Image Annotator browser interface

4 Methods

4.1 Baseline

For generating baseline results, we decided to observe the predictions of the existing models on our dataset. For this purpose, we used the model from SpotGarbage paper and existing Mask RCNN model from Facebook which was trained on COCO dataset.

- SpotGarbage model: The original code and model is in Caffe. We wanted to get exposure to the PyTorch deep learning framework. We used an open source tool called Model Management DNN(MMDNN), developed by Microsoft to convert neural network architectures from one framework to another. After converting and modifying the code provided by the SpotGarbage paper to work on the PyTorch framework, we obtained bounding boxes on images (Figure 5).
  - The generated bounding boxes had irregular shapes and had multiple objects inside them. This is undesirable for our task of extraction.

Figure 5: Baseline from GarbNet after PyTorch coversion

- Detectron 2: We used the existing open source code for generating instance segments on images from our dataset.
  - The segments and bounding box generated by the model are not at all appropriate. It has no sense of what garbage is and is generating completely wrong predictions as can be observed from Figure 6

4.2 Mask R-CNN

The baseline only gave us bounding boxes and as shown in Figure 5, the bounding boxes aren't always very accurate in depicting where the garbage is. In Figure 5, we can see that the baseline model gives us three bounding boxes instead of one. Not only does it perform poorly, but we would not be able to send a robot to pick the garbage up since it is not very clear where it is.

The difference between what the baseline model is doing and our goal is the following: As we see in Figure 7 below, the bottom right image is an example of instance segmentation of balloons. Instance segmentation identifies not only the regions of the class but also each instance of the class. Apart from creating bounding boxes over these regions, it creates a unique color mask for each instance of the class.

This is why we decided to go with a mask regional convolution neural network (Mask R-CNN) architecture. The way Mask R-CNN works is a two step process.

Figure 6: Baseline from Mask RCNN (Detectron2)

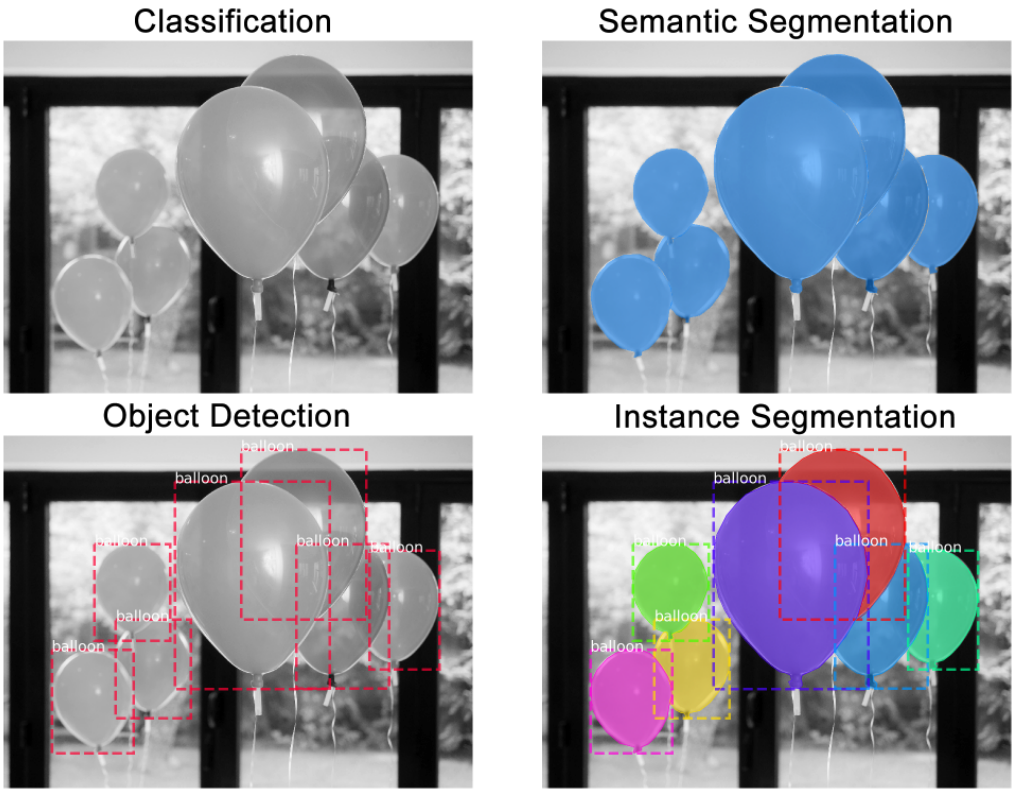

Figure 7: Difference in Instance Segmentation

1. Provide refined regions of interest and classify the region.

2. Create a mask over the classified portion of the image.

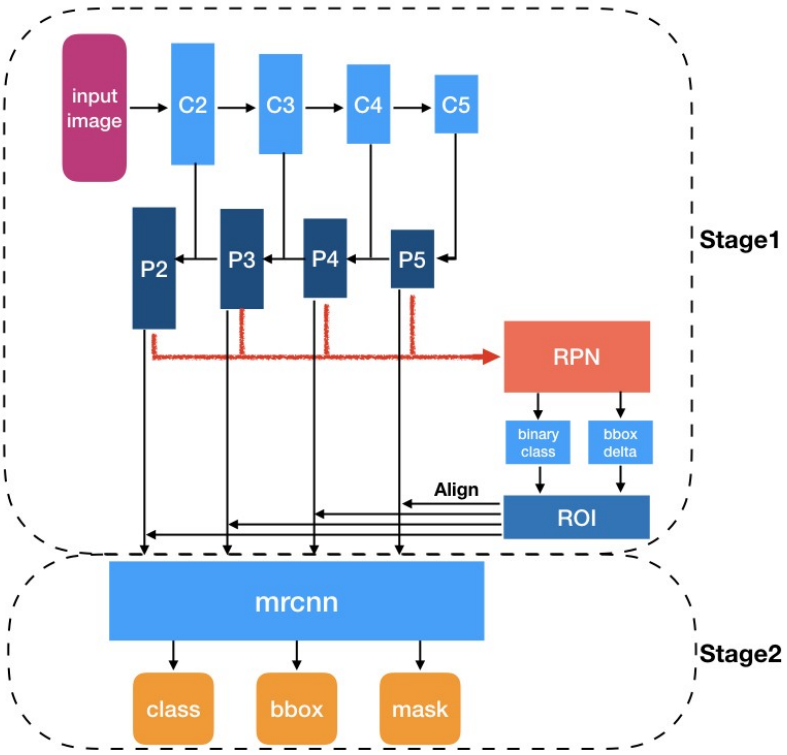

Figure 8: Xiang Zhang illustration of Mask R-CNN

The architecture of this network considers several regions of the image and then refines those proposed regions. Several layers of convolutions are used to extract features from the images and are passed through more convolutions in a pyramid-like structure (convolutions C2-C5 in Figure 8). These are then pooled and reverse pooled for the mask as well as the region proposal network (RPN in Figure 8 during stage 1).

The RPN first refines the region then classifies the region and in parallel creates a bounding box around the region of interest. This is then outputted as a region of interest (ROI in Figure 8 during stage 1).

This ROI is fed into the mask network to output the bounding box, the pixels associated with the classification and the class prediction.

The class is outputted for multi-instance segmentation where the network could identify several different classes in an image. For this project, we only considered the garbage as the class but in the future, we look to expand the dataset to contain several different classes such as glass, plastic bottles, metals, etc. for recycling.

4.3 Training OscarNet

Training our model from scratch was not a possibility with limited resources and computation power. We used transfer learning to overcome this barrier. We used a pre-trained Mask RCNN model. It was trained on the COCO dataset, a huge dataset of common objects in context including pets, people, vehicles, etc. As a result, our model already had basic understanding of lines, curves and edges. So, while backpropagating we didn't have to propagate all the way back to the first layer.

OscarNet was trained on an NVIDIA GPU with 8 GB of memory. Training for 30 epochs took us around 15-20 hours and each epoch took approximately 30-45 minutes. OscarNet was trained on 150 images. We had a small validation dataset of 10 images and the remaining 24 images were for testing.

4.4 Bit Mask Segmentation

We take the original image and convert it to greyscale. We then take the mask output from OscarNet and superimpose the classified pixels onto the grayscale image, leaving the classified pixels in color. This creates a colored segmentation over the classified region while everything else is in greyscale. Along with this, we also implemented the code for displaying a bounding box over the generated masks.

The final result is shown below where Hunter, our team member, posed with a garbage bag in one hand and a piece of crumpled plastic bag in the other. As described, OscarNet's predicted masks are left in color while everything else is in greyscale.

Figure 9: Team member Hunter Hobbs posing for OscarNet. Original (left), prediction (right).

4.5 Image Depth Estimation

For the robot to navigate towards the detected trash, it needs to be aware of its surroundings and the depth of that segment. To get the depth estimate of a pixel, we used MonoDepth2(3). We used a pre-trained MonoDepth2 model which takes in monocular (single pane) images and predicts the pixel depth in the image creating a heat-map like an image where the closer objects are shown in brighter colors. In order to get the actual depth in meters, we need the baseline depth of the Depth Estimation model(0.54m in this case) and the focal length of the camera being used. Since the focal length varies from camera to camera, we decided to use Intel's Realsense camera which generates a depth map and can be used to calculate the depth.

4.6 Experimental setup and Pipeline

Now that we have the ability to detect garbage and its depth, we created a pipeline to automate the process. We set up a system that would take images using the realsense camera every 60 seconds and run both OscarNet and MonoDepth2 on it. We then generated directional commands to move to the closest garbage segment in the image. For example, for the image below Figure 12, our system generated a command like "Travel left at an angle of 50 degrees for 0.3m"

Figure 10: Before and after image results of MonoDepth2

Figure 11: Pipeline setup

4.7 Dispatch the Robot

The directional commands we generated are generic and can be used on any bot or quadcopter to reach the closest garbage according to its frame of reference. We mapped these commands for Sparki, an easy bot by Arcbotics. We designed another experimental setup where we mount the camera at some height from the ground and keep taking photos of a fixed frame every 60 seconds. We then threw some garbage in that frame and our pipeline generated commands for locomotion to that garbage. We then dispatched Sparki to that location. Since the bot doesn't have a Bluetooth chip, we couldn't automate the complete process of picking up the trash. We had to connect the bot to our computer and send the commands over. We plan to automate this using a different bot as our future work.

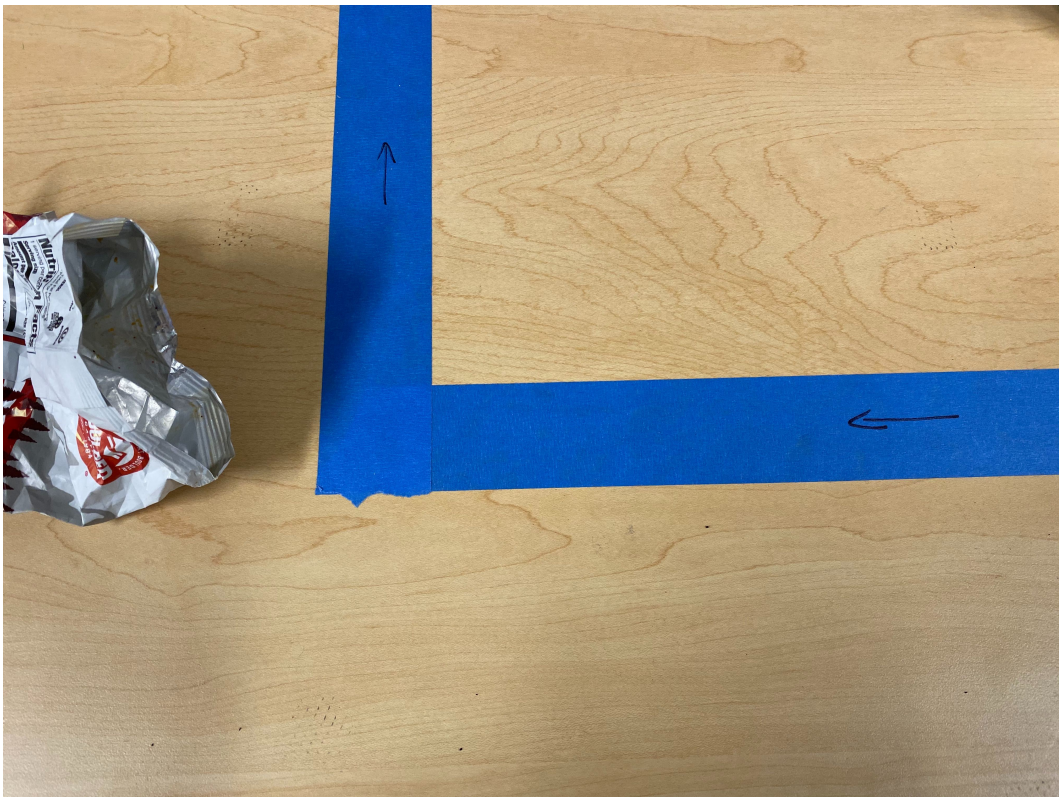

Figure 12:

5 Results

Some of the results can be seen from figures below: Figure 13, 14, 15 and 16

Object detectors on COCO(2) generally use 12 metrics for evaluation. Three of them are $AP^{IOU=0.5:0.05:0.95}$, $AP^{IOU=0.5}$, $AP^{IOU=0.75}$. For our dataset, we just have one class, and so by definition mAP and AP are the same. We have calculated the above three metrics which can be seen in Figure 19, Figure 20 and Figure 21.

Figure 17 has values for various different losses of two models that we observed on our training data.

5.1 Extraction Results

The initial test setup was a result of using an off-the-shelf robot named Sparki developed by Arcbotics and a stand alone camera. The camera was set at a predetermined distance proportional to the focal length of the lens. The position of the camera was such that it was taking pictures of the floor. The camera gave a live stream where a frame was processed each 60 seconds with OscarNet to determine if there was garbage. Once garbage was placed in the frame and OscarNet detected it, MonoDepth2 can determine the distance. Sparki was set to an origin point in the frame of the camera and given instructions on the angle of pivot and distance to cover in order to meet the trash.

5.2 Performance Metrics

To quantitatively asses the performance of the system we used the mean average precision (mAP). In order to consider if the classified region is correct we used Intersection over Union (IoU). IoU is a threshold which takes into account the overlap of the proposed classification region and the true region as well as the total area covered by both regions. If the ratio is 1 then the classification is perfect, if it is 0 then the classification is completely wrong. A common threshold to set for a

Figure 13: Identifying the mask and drawing a bounding box around identified garbage

correct classification is 50% but we experimented with several threshold values from 50% to 100% increasing by 5%.

Mean average precision is calculated by taking the average precision of the classifier and then taking the mean of that. Below we see the results of the these metrics compared the the pre-trained model which was trained on the COCO dataset.

Intuitively as we increase the threshold we are only considering ROI that are tighter to the ground truth as correct classifications leading to poorer performance. If we had more annotated data and more time to train the model it would do better at being tighter to the ground truth region.

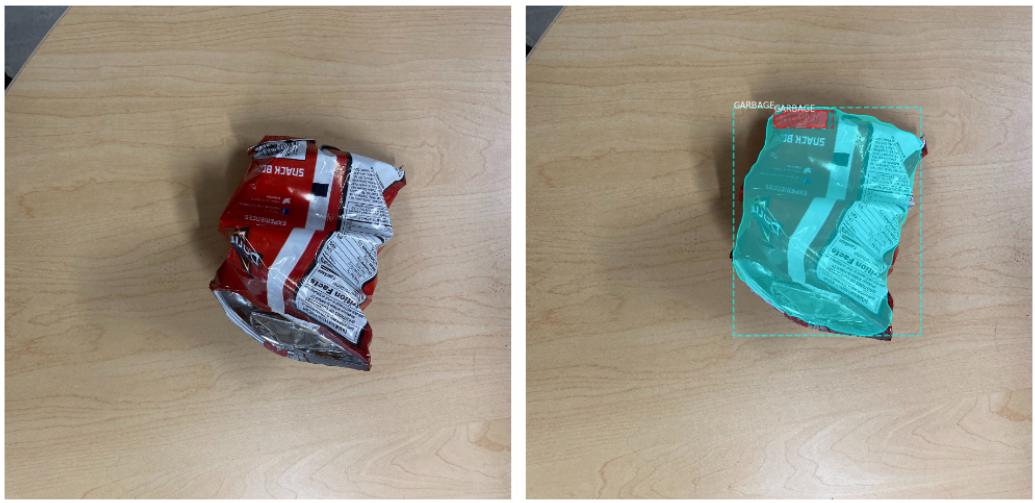

Figure 14: Identifying the mask and drawing a bounding box around identified garbage

Figure 15: Identifying the mask and drawing a bounding box around identified garbage

Figure 16: Identifying garbage in a multi-object scenario. As you can see there are no false positives.

LossType	*COCODatasetModel*	*SSGarbageDatasetModel*
loss	3.5599	0.2051
rpnClassLoss	0.0313	0.0063
rpnBboxLoss	0.4920	0.0723
mrcnnClassLoss	0.9339	0.0063
mrcnnBboxLoss	0.9532	0.0165
mrcnnMaskLoss	1.1495	0.1036

Figure 17: Loss values

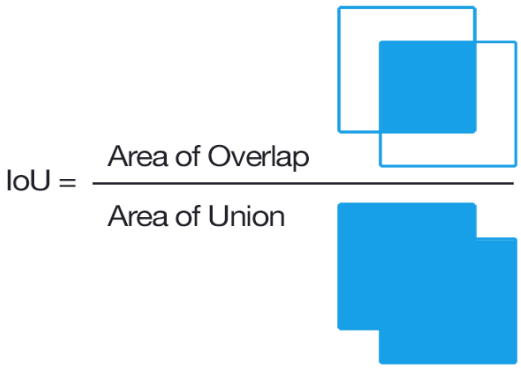

Figure 18: Intersection over Union threshold

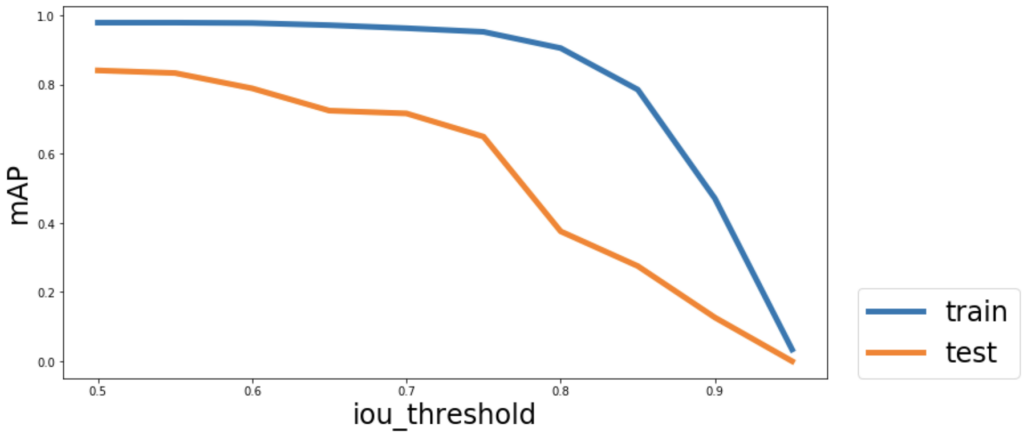

Figure 19: Plot of mAP with IoU values

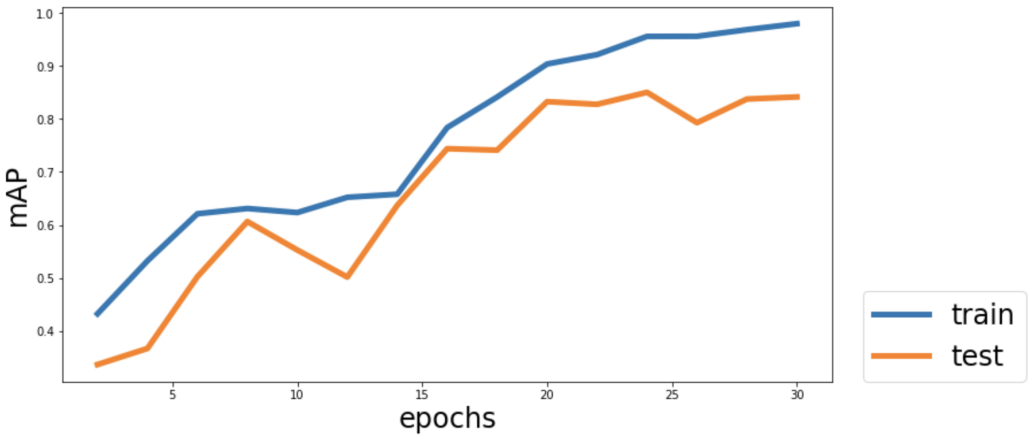

Figure 20: Plot of model improvement with number of epochs

$Model$	$Dataset$	$iou\_threshold$	mAP_train	mAP_test
$MR-CNNBaseline$	$COCO$	0.5	0.432	0.336
$MR-CNNBaseline$	$COCO$	0.75	0.216	0.093
$ModifiedMR-CNNTrain$	$SSGarbage$	0.5	0.980	0.842
$ModifiedMR-CNNTrain$	$SSGarbage$	0.75	0.953	0.650

Figure 21: Pre-Trained model vs. OscarNet on different IOU thresholds

6 Conclusions

In conclusion, we created an autonomous system to detect and remove garbage. This involves a classifier that does instance segmentation as the first step following which we do a depth estimation to exact the location of the garbage. We generate generic commands to guide a bot to complete the task of picking up the garbage.

To achieve the above goal, we created our dataset called SSGarbageNet and used Mask R-CNN with transfer learning from a pre-trained model to train OscarNet. We reached an mAP of 98% on the training data and 84% on the validation data with an IoU threshold of 0.5. We simulated this successfully by sending Sparki the robot navigation commands after depth estimation to reach the garbage.

7 Future Work

In future work we plan to:

1. Expand our dataset to contain different classes of garbage such as recyclables, paper, hazardous waste, etc.
2. Increase training time on a more expansive dataset
3. Use higher performance GPUs for training
4. Mount a camera directly onto Sparki for completely autonomous task execution.
5. Path finding on Sparki with added obstacle avoidance
6. Generate an optimal path for garbage cleaning

References

[1] Emily Holden. This article is more than 5 months old us produces far more waste and recycles far less of it than other developed countries, 2019. URL `https://www.theguardian.com/us-news/2019/jul/02/us-plastic-waste-recycling`.

[2] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[3] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. October 2019.

[4] M Garg N C Krishnan G Mittal, K B Yagnik. Spot garbage: Smartphone app to detect garbage using deep learning. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '16, pages 940–945, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4461-6.

[5] Xu Zhang Ying Wang. Autonomous garbage detection for intelligent urban management. In *Matec Web of Conferences*, 2018 2nd International Conference on Electronic Information Technology and Computer Engineering (EITCE 2018), Paris, France, 2018. EDP Sciences. doi: https://doi.org/10.1051/matecconf/201823201056.

[6] Ying Liu, Zhishan Ge, Guoyun Lv, and Shikai Wang. Research on automatic garbage detection system based on deep learning and narrowband internet of things. *Journal of Physics: Conference Series*, 1069:012032, aug 2018. doi: 10.1088/1742-6596/1069/1/012032. URL `https://doi.org/10.1088%2F1742-6596%2F1069%2F1%2F012032`.

[7] Shuqiang Jiang Xinhang Song, Luis Herranz. Depth cnns for rgb-d scene recognition: Learning from scratch better than transferring from rgb-cnns. In *AAAI*, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), pages 4271–4277, 2017.

[8] Abhishek Dutta and Andrew Zisserman. The via annotation software for images, audio and video. In *ACM International Conference on Multimedia*, volume 27 of *MM 19*, New York, USA, Oct 2019. ACM, ACM. doi: 10.1145/3343031.3350535. to appear in Proceedings of the 27th ACM International Conference on Multimedia (MM 19).