CS230

Waste Object Detection and Classification

Hrushikesh N. Kulkarni
Department of Computer Science
Stanford University
hrushi@stanford.edu

Nandini Kannamangalam Sundara Raman
Department of Computer Science
Stanford University
nandini9@stanford.edu

Abstract

We try to categorize the different pieces of the waste in an collaged image into categories : glass, plastic, paper, trash, metal, cardboard. We have tried Hybrid Transfer Learning for Classification and Faster R-CNN to get region proposals for object detection. First, we define our waste classification problem and present current research and solutions to similar object detection problems. We then give an outline of our proposed architecture for creating collages using GANs and model for approaching our specific task object detection, which uses a fine-tuned Faster R-CNN. We will also describe the nature and generation of our dataset, as well as the results we achieved from our experiments. Lastly, we outline our next steps in terms of optimizing and improving upon our solution.

1 Introduction

Improper waste management has severe effects on our environment, natural resources and public health. We want to educate users to be more mindful of recycling throwaway items, so that we can reduce the contamination at the source. We use pictures of different throwaway items from Yang and Thung (2016) and train a classifier to output a label which is a type of the object. Knowing the labels, the user can make a sound decision if it has to be recycled.

In our report we discuss innovative ways to augment the dataset, and present a object detection mechanism which as been trained on different throwaway classes. We also experiment different hyper-parameters for these learning algorithm and present a summary.

2 Related work

Object detection and classification approaches for throwaway items is a well studied topic. Yang and Thung and Chu et al. use a AlexNet Krizhevsky et al. (2012) like architectures, and have very poor accuracy. Yang and Thung (2016), their classifier was confused between plastic and glass categories.

To have more robust classification we experiment with different classifiers namely ResNet by He et al.. Because of the skip connection mechanism in He et al. (2016) we find that the ResNet worked the best.

3 Dataset and Features

We use TrashNet (Yang and Thung, 2016) dataset as the baseline. This is a balanced dataset and has approx 400 images in each of 6 different labelled classes (Glass, Paper, Cardboard, Plastic, Metal, Trash). We divide the dataset into train/validation/test as (80/10/10). The resolution of the input data is $512 \times 384 \times 3$.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

(a) Trash (b) Paper (c) Metal (d) Glass (e) Cardboard (f) Plastic

Figure 1: Sample Training Set from the TrashNet Dataset Yang and Thung (2016)

We augment this dataset by adding operations like flipping, rotation, sheering, etc. Since the dataset has only one object in a image, we augment this by making a array of collage as mentioned in Sec. 3.1.

3.1 Multiple Objects in a Image

To increase the quantity of the dataset, we generate a collage of multiple of these input images. We present three methods of generating collages below.

- Placing images at random
- Placing images at 4 quadrants
- Learning where to place the images

3.1.1 Placing images at random

This dataset has images cropped by hand, and randomly placed on a white canvas. The objects on these images overlap each other. We have around 2000 such images along with the bounding boxes of the objects.

3.1.2 Placing images at 4 quadrants

This dataset has images cropped by hand, and placed on 4 quardrants of a white canvas. The objects on these images do not overlap each other. We have around 2000 such images along with the bounding boxes of the objects.

(a) (b) (c) (d)

Figure 2: Collages by randomly placing images (a) and (b) and placing images at quadrants (c) and (d)

3.1.3 Learned Collage

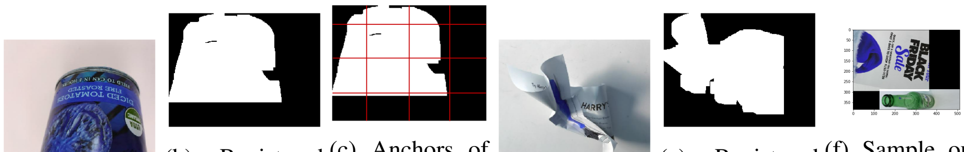
We pose generating collages as a Knapsack problem Wikipedia (2019) optimization problem learning to find optimal placement of two images such that they have minimum overlap.

Algorithm 1: Registering images

Two images A and B
Find the respective foreground masks A_{mask} and B_{mask}
Translate A_{mask} and B_{mask} such that
$B_A = Boundingbox(A_{mask})$
$B_B = Boundingbox(B_{mask})$
$Topleft(B_A) = (0, 0)$
$Topleft(B_B) = (0, 0)$

After registering all images, we convolve B_{mask} over A_{mask} and find the collage loss as given in Eq. (2) at every location.Now by translating B_{mask} at each anchor point, we can find the anchor loss at each anchor point as given in 4.

$$\begin{bmatrix} x'_{B_{mask}} \\ y'_{B_{mask}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \begin{bmatrix} x_{B_{mask}} \\ y_{B_{mask}} \\ 1 \end{bmatrix} \tag{1}$$



(a) Image A (b) Registered A_{mask} (c) Anchors of A_{mask} (d) Image B (e) Registered B_{mask} (f) Sample output

Figure 3: Image registration and anchor points

Now we define a collage loss, so that we have minimum overlap and maximum union area between the foreground.

$$\begin{aligned} intersection &= A_{mask} \bigcap B_{mask} \\ union &= A_{mask} \bigcup B_{mask} \\ collage_{loss} &= \frac{intersection}{union} \\ anchor_{loss} &= [l_1, l_2, l_3, \ldots] \end{aligned} \tag{2}$$

We collect the ground truth by exhaustively searching the anchor points for all the images in dataset for A and B.

$$B_{anchor} = argmin(anchor_{loss}) \tag{3}$$

We use the median of all such points to indicate the location of optimal placement with respect to image A

$$A_{optimal} = median\,[B_{anchor}, C_{anchor}, D_{anchor}, \ldots] \tag{4}$$

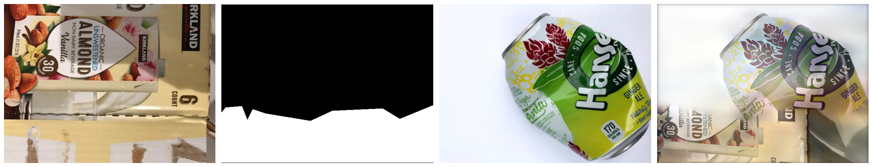
To resize the image at first, we start with a max pooling layer, and then followed by a LeNet like architecture. This approach was promising at first, it did not work for the following reasons.

This is a balanced dataset and has approx 4Using all registered images, cause the output anchor point to be always the same. Hence all generated collages looked very similar. It took too much time to calculate the ground truth.

3.1.4 Blending Images using GP-GAN

To blend different masks on a collage we experimented with a pre-trained GP-GAN Wu et al. (2019). This approach uses Gaussian-Poisson Generative Adversarial Network (GP-GAN), which poses the

Figure 4: CNN for learning anchor points



(a) Image A (b) Mask (c) Image B (d) Result

Figure 5: Using GP-GAN Wu et al. (2019) to blend images A and B using a Mask.

joint optimization between the gradient and image color information. We used pretrained weights available from Wu et al. (2019), and generate a sample output in **??** . We decided not to go in favour of this approach, since it blurs image features we suspected that it will hurt the performance.

4 Methods

In this section, we describe different ways to generate a collage, and methods for classifying and localizing objects in a image.

4.1 Hybrid Transfer Learning

We use a Hybrid approach to Transfer Learning described in Géron (2019). For trying this out, we start with pretrained models on image net. Then add a Global Average Pooling layer, followed by Batch Normalization, then Dense output layer. We start by freezing all base layers, and training the output layer by a large learning rate, then after a few epochs, we reduce the learning rate drastically. The advantage of this approach is that it does not change the weights of the base layers drastically in the first few epochs when the last few layers have not yet stabilized.

4.2 Fine-tune with Faster R-CNN network

Faster R-CNN has a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thereby allowing nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection.

We used Faster R-CNN (Fig.7a) network with Inception V2 trained on MSCOCO dataset as the baseline and conducted experiments by fine tuning it with TrashNet dataset with hand annotated bounding boxes, randomly generated collages (3.1.1), collages with images at 4 quadrants (3.1.2) and their bounding boxes.

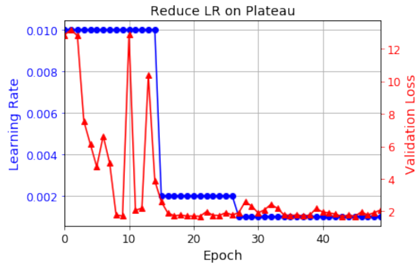
5 Experiments/Results/Discussion

5.1 Hybrid Transfer Learning for Classification

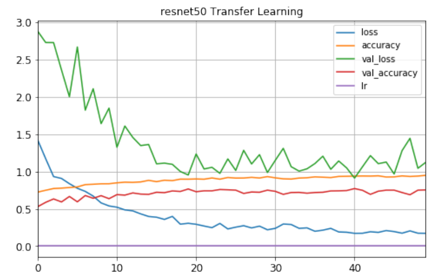
- Reduce Learning Rate on Plateau. We reduce the learning rate whenever the validation score reaches a plateau. (Fig.6b)
- Learning Rate Schedule

We try different learning rates for 4.1.

	$lr_1 = 0.01, lr_2 = 0.01$	$lr_1 = 0.001, lr_2 = 0.001$	$lr_1 = 0.01, lr_2 = 0.0001$
Precision	0.15	0.06	0.18
Recall	0.91	0.16	0.19
F1-Score	0.10	0.07	0.18



(a) Reduce the learning rate by 0.2



(b) Hybrid Transfer Learning with $lr_1 = 0.01$, $lr_2 = 0.0001$

Figure 6: Plot showing the learning rate schedule, and training characteristics

The confusion matrix for Hybrid Transfer Learning is below

	cardboard	glass	metal	paper	plastic	trash
cardboard	**10**	2	4	9	2	8
glass	1	**6**	7	9	5	7
metal	6	7	**5**	8	4	8
paper	7	8	7	**6**	3	10
plastic	8	9	6	8	**4**	14
trash	4	6	9	7	4	**16**

5.2 Fine-tune with Faster R-CNN network for Object Detection

Different learning rates were tried for fine-tuning object detection network with TrashNet Collage (3.1.2) with a minibatch size of 16 using baseline as Faster RCNN on Inception network V2 trained on MSCOCO and we found that learning rate of 0.0002 gives us the best. Precision/Recall and F1 scores for object detection

LR	0.0002 (TrashNet)	0.0002 (Collage)	0.00002 (Collage)	0.000002 (Collage)
Loss	0.436	1.214	1.442	1.452
Precision	0.816	**0.842**	0.697	0.7
Recall	0.565	**0.878**	0.791	0.793
F1-Score	0.668	**0.859**	0.708	0.743

	cardboard	glass	metal	paper	plastic	trash
Precision	0.97	0.76	0.81	0.90	0.84	0.77
Recall	0.99	0.82	0.86	0.90	0.87	0.8
F1-Score	0.98	0.79	0.83	0.9	0.85	0.78

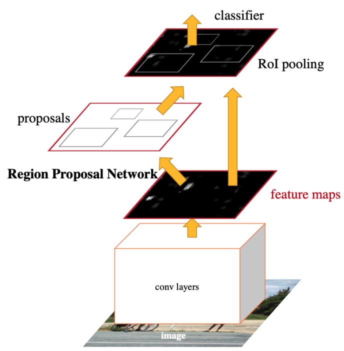
Metrics for cardboard seems to be the highest and next followed by paper. This maybe due to the uniqueness (cardboard has a brown texture and paper has printed material on it) of the textures of the objects of these two classes in the dataset compared to other objects.

6 Conclusion/Future Work

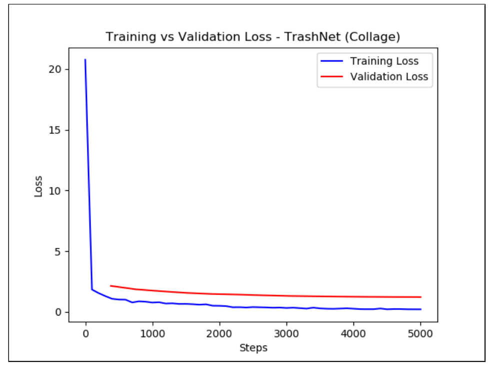
The Hybrid training requires two learning rates, and is harder to train. We also choose not to use GP-GANs because the GAN blending blurs the features of the image,and hurts the performance.

Fine tuned Faster R-CNN returned good object detection results for learning rate 0.0002.

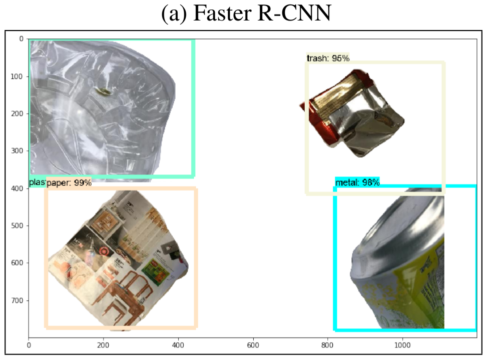
Given that we can generate infinite number of collages with the original dataset, we can generate sufficient data to train the model from scratch. With time permitting, we can train the model from scratch and use that instead of the pre-trained model. We also hope to test our model on real images of piles of trash in order to determine how our model does on the actual problem.



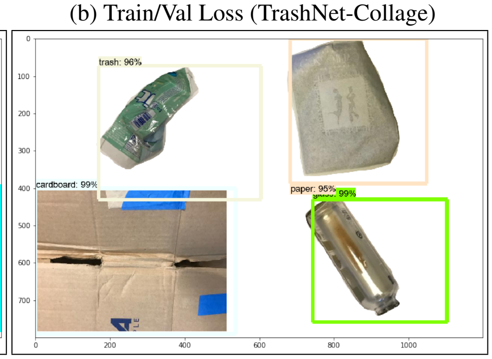
(a) Faster R-CNN



(b) Train/Val Loss (TrashNet-Collage)



(c) Object Detection Results



(d) Object Detection Results

Figure 7: Train Validation Loss and Object Detection Results for Fine-tuning Faster RCNN with TrashNet collages

7 Contributions

Hrushikesh worked on Learning Collages, blending images using GP-GAN, and classification using Hybrid Transfer Learning.

For Milestone 1, Nandini worked on reproducing the baseline model for classification problem and object detection on randomly collaged images created by her. Then she worked on creating collages in 4 quadrants and Fine-Tune Faster R-CNN for object detection using TrashNet dataset and TrashNet Collages.

References

Chu, Y., Huang, C., Xie, X., Tan, B., Kamal, S., and Xiong, X. (2018). Multilayer hybrid deep-learning method for waste classification and recycling. *Computational Intelligence and Neuroscience*, 2018.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Wikipedia (2019). Knapsack problem. https://en.wikipedia.org/wiki/Knapsack$_{p}roblem.$

Wu, H., Zheng, S., Zhang, J., and Huang, K. (2019). Gp-gan: Towards realistic high-resolution image blending. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2487–2495. ACM.

Yang, M. and Thung, G. (2016). Classification of trash for recyclability status. *CS229 Project Report*, 2016.