Computer Vision Assisted Approaches to Detect Street Garbage from Citizen Generated Imagery

Hye Seon Yi(✉) and Sriram Chellappan

University of South Florida, Tampa, FL 33620, USA
{hsyi,sriramc}@usf.edu

Abstract. The basis of smart governance is to leverage state-of-the-art technologies to improve lives of citizens. With the rapid permeance of smart-phone technologies today, citizens are increasingly active now in collaborating with public officials for improved quality of life. However, for effective utility, public officials must be empowered with optimal tools that can best leverage citizen participation. In this paper, we present the design and details of computer vision techniques to automatically detect and localize street garbage from citizen generated imagery, and analyze the performance of multiple techniques. Our dataset is mined from (citizen-generated) images in the well-known 311 service deployed in San Francisco, which is actually a service citizens use to report civic issues. Using a dataset of 2,500 images (containing 6,474 objects) evenly distributed between those containing street garbage and those that do not, we design and compare convolutional neural network techniques to detect and localize sources of garbage in the images. Results from our evaluations show that our system can be a vital cog towards next generation smart governance systems geared towards cleaner and healthier neighborhoods. Since identifying, collecting and disposing of street garbage is a critical aspect of governance across the globe, we believe that our work in this paper is critical, timely and may have global impact.

Keywords: Object detection · Garbage detection · Public health · Smart governance · Transfer learning · Computer vision

1 Introduction

Across the globe, there are urgent efforts now to rethink governance from the ground up to tackle various challenges including rising populations, keeping them healthy, combating climate change, managing rising floods, ensuring availability of food and water, providing education, and so much more. Unfortunately, the challenges are only mounting. In this context, and especially with the ubiquity and affordability of smartphones and network connectivity, citizens are now increasingly able to support local governance efforts. Furthermore, with the advent of social media, most gaps between officials and citizens are only

S. Paiva et al. (Eds.): SmartCity360° 2020, LNICST 372, pp. 526–541, 2021.
https://doi.org/10.1007/978-3-030-76063-2_35

shrinking even further. There is hence a rich set of emerging literature on leveraging citizen generated data for improving governance efforts across many fronts including water management, public health, law enforcement, intelligent transportation and much more.

In this context, a critical service provided by governmental agencies across the globe, is keeping localities cleaner and free from garbage. Needless to say, excess or abandoned garbage has serious repercussions to society today including attracting criminals, attracting pests and dangerous animals, low property assessments, contaminated soil, foul odors and so much more. Unfortunately, despite best efforts, there are always sources of abandoned garbage even in high-income countries today, and the problem is much worse in medium and low income countries. In this paper, we present the design and details of using Artificial Intelligence (Computer Vision) techniques to provide an automated mechanism to detect and localize multiple sources of garbage (cardboard boxes, loose garbage, garbage can and garbage can overflow) from images generate by citizens themselves.

Table 1. Communication preferences of 311 requests among citizens in San Francisco

Communication channel	311 Requests	With image	No image
Phone	1,777,585	6,279	1,771,306
Mobile/Open311	1,213,300	952,014	261,286
Website	564,849	38,266	526,583
Third party agency	123,907	2	123,905
Twitter	31,776	8,101	23,675
Other	6,953	6	6,947
Total	3,718,370	1,004,668	2,713,702

The dataset for our problem is generated from publicly available 311 services that citizens in San Francisco [1] use to report civic problems. While, we present more details later, 311 services are available at most big US cities [2], and serves as the primarily customer service center for civic problems. In San Francisco, the service is available via phone calls, a mobile app, a dedicated website, and Twitter. Citizens can call to report problems, and also use the app, website and Twitter to do the same, while also uploading picture of problems they see. As of Aug 2019 in San Francisco alone, there are more than 3.7 million citizen generated civic reports, and there are more than 1 million images that citizens have uploaded so far. Table 1 presents details of 311 use in San Francisco alone, from where we see that Internet based platforms (app, website, Twitter) are very popular among citizens. From this dataset, we utilized 2,500 images (containing 6,474 objects) for the current study focusing specifically on designing computer vision techniques for automatic identification and localization of garbage within an image.

We design, analyze and compare two pre-trained CNN models for our problem in this paper. The techniques are a) Faster R-CNN [3] and b) RetinaNet [4], both of which are successful object detection models with high performance [5]. Basically, in Faster R-CNN technique, feature maps are extracted from our training images using convolutional neural networks. The class to be learnt for our problem comprises of cardboard boxes on roads, loose garbage, garbage cans and overflowing garbage cans. Subsequently, and in order to localize objects of interest in an image, we train a simple neural network to learn features embedded within annotated objects of interest in the training images, and from the feature maps derived earlier in the previous step. Subsequently, these steps are repeated during the testing phase, wherein, once an image comes in, our model will identify and report either a) no garbage class is present; or b) garbage class is present, and also localize where the detected object of interest is present in the image via bounding boxes. RetinaNet on the other hand is a single network that is comprised of a backbone network and two task-specific sub-networks, The backbone network computes a convolutional feature map over an entire input image and is an off-the-self CNN. The first sub-network performs classification on the output of the backbone, while the second sub-network performs convolution bounding box regression.

Table 2. Categorical columns sorted by 311 requests with images.

#	Service_name	Service_subtype	Service_details	Requests	Percent
1	Street and Sidewalk Cleaning	general_cleaning	other_loose_garbage	177,897	17.82%
2	Encampments	encampment_reports	encampment_cleanup	88,822	8.90%
3	Street and Sidewalk Cleaning	bulky_items	furniture	57,364	5.75%
4	Street and Sidewalk Cleaning	human_or_animal _waste	human_or_animal _waste	48,356	4.84%
5	Street and Sidewalk Cleaning	bulky_items	boxed_or_bagged _items	42,341	4.24%
6	Abandoned Vehicle	abandoned_vehicles	dpt_abandoned _vehicles_low	30,168	3.02%
7	Graffiti	graffiti_on_other _enter_additional _details_below	other_enter _additional_details _below_offensive	27,836	2.79%
8	Street and Sidewalk Cleaning	bulky_items	mattress	22,020	2.21%
9	Graffiti	graffiti_on_building _commercial	building_commercial _not_offensive	21,163	2.12%
10	Street and Sidewalk Cleaning	city_garbage_can _overflowing	city_garbage_can _overflowing	21,056	2.11%
11	Illegal postings	illegal_postings _affixed_improperly	affixed_improperly	19,088	1.91%

To the best of our knowledge, we are not aware of any study that specifically focuses on detecting and localizing garbage from citizen generated imagery. We believe that our work is practical, and can be an important tool for next generation smart and automated governance systems across the globe (especially as it pertains to detection of a variety of garbage sources). The comparisons we do between the two CNN techniques we employ in this paper are expected to benefit policy-makers in real-time when attempting to use AI techniques for citizen science applications.

2 Our Dataset Consisting of 311 Images

The 311 service is popular in large cities in the US, wherein citizens can report civic related issues and complaints for rapid addressing. Across the US, these services are available to citizens via phone, websites, apps, and sometimes through third party web based intermediaries. When a citizen contacts 311 via any medium, it creates a 311 request. These requests are all available to the public. For the specific case of San Francisco - one of the biggest cities in the US - the information on all reports is available on DataSF, a San Francisco open data portal (https://datasf.org/opendata/). DataSF uses Socrata [6] platform for their data management, which provides APIs to access its data programmatically.

Using the Socrata APIs, we were able to collect the total of $3,718,370$ requests dated from July 1, 2008 to August 28, 2019. Among those requests, there were $1,004,668$ requests that had images uploaded by citizens. Tables 2, 3 and 4 present details on the dataset. Table 2 presents details of various requests in San Francisco with images from citizens, categorized by type of service requested. Table 3 provides a simpler to visualize categorization of the same, and finally, Table 4 provides details on the dataset for our problem in this study, that comprises of $6,474$ objects (in $2,500$ images) in total, out of which $5,745$ were used for designing our AI models, and 729 were used for validating and testing the AI models. Figures 1 and 2 shows a snapshot of images in our dataset for clarity.

Table 3. Service names sorted by 311 requests with images.

#	Service_name	Requests	Percent
1	Street and Sidewalk Cleaning	423,485	42.15%
2	Graffiti	219,384	21.84%
3	Encampments	96,417	9.60%
4	Parking enforcement	41,241	4.10%
5	Abandoned vehicle	30,313	3.02%
6	Illegal postings	26,482	2.64%
7	Sign repair	20,777	2.07%

Table 4. Number of objects used in the train and test datasets.

Class	Total	Training (%)	Validation and Testing (%)
cardboard_box	1,490	1,319 (89%)	171 (11%)
garbage	3,244	2,856 (88%)	388 (12%)
garbage_can	864	775 (90%)	89 (10%)
garbage_can_overflow	876	795 (91%)	81 (9%)
Total	6,474	5,745 (89%)	729 (11%)

Fig. 1. Examples of garbage and cardboard boxes in our dataset

3 Our Proposed Methodology for Object (Garbage) Detection and Localization

For the purposes of this study, we chose four classes, namely "garbage", "garbage_can", "garbage_can_overflow", and "cardboard_box", for object detection. Note that these were all categorized as "Street and Sidewalk Cleaning" in 311 dataset which was the most requested service with images in San Francisco. Table 4 shows details of number of images in our dataset.

3.1 Images and Labeling

After deciding the four classes for object detection, we need to create ground truth data for the study. First, we downloaded related images containing those four class objects. We picked out about 2,500 images for object detection, that contained 6,474 objects of interest (in Table 4), since in many images more than one object of interest was present. Then, we labeled those images manually using LabelImg, which is one of the widely used image annotation tools for labeling images [7]. Figure 3 is the screenshot of LabelImg. Essentially, the task here is manually emplacing bounding boxes around the object of interest, so that the algorithm designed learns to detect the object of interest (if present) in an image, and also localize it by emplacing bounding boxes. In this manner, even minor

Fig. 2. Examples of garbage cans and overflowing garbage cans in our dataset

garbage boxes could be detected better from the perspective of a human operator that is viewing these images in run-time.

We divided these 2,500 labeled ground truth images with 6,474 objects for training, validation and testing datasets. For training, we used 90% and for validation and testing, we used 10% of images [8]. Table 4 presents details of training, validation and testing.

3.2 Transfer Learning with Pre-trained Object Detection Models

Many of the current state-of-the-art object detection models are based on deep convolutional neural networks (CNN) which are one of the commonly used deep learning methods [9–11]. CNNs can automatically extract meaningful features from images whereas traditional object detection models used feature-based and statistical machine learning methods [12]. However, training CNN models from scratch is not an easy task because it requires a lot of labeled data and computing power. Moreover, creating labeled image data is time consuming and expensive because each image has to be examined and labeled manually. To remedy this problem, transfer learning is a highly recommended alternate approach. The idea of transfer learning is to train a model using the knowledge learned from excellent pre-trained models, that work for a broad class of problems. These pre-trained models were trained using large existing labeled image dataset, such as ImageNet [13] and KITTI [14]. In this study, we used two pre-trained object detection models namely Faster R-CNN and RetinaNet.

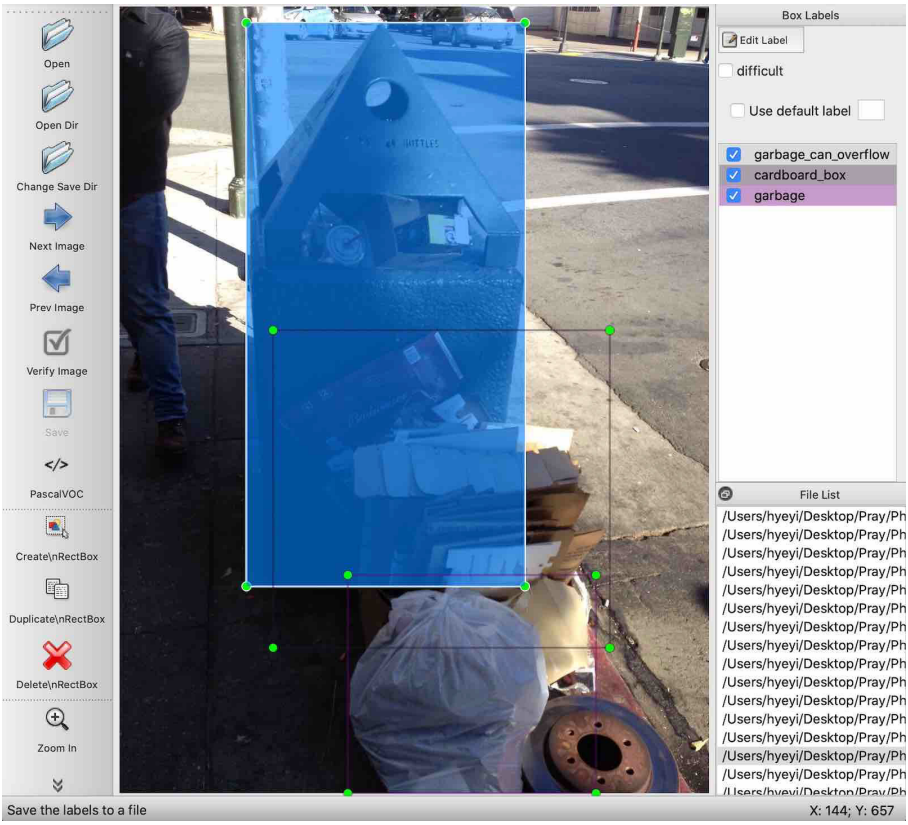
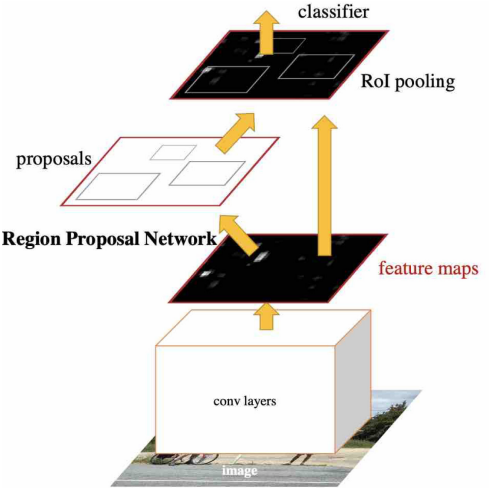
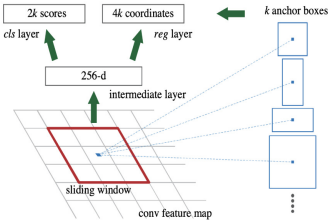


Fig. 3. Screenshot of LabelImg

Faster R-CNN. Faster R-CNN proposed by Ren [3] is one of well-known and successful object detection models [12,15]. This model accomplished the highest accuracy on PASCAL VOC in 2007 and 2012 and the models based on Faster R-CNN won 1st place in several tracks in ILSVRC and COCO competitions in 2015 [16]. [3] describes Faster R-CNN as follows. Faster R-CNN is composed of two modules: a Region Proposal Network (RPN) and the Fast R-CNN detector [17]. A RPN is a fully convolutional network which proposes regions. Fast R-CNN is a predecessor of Faster R-CNN and the Fast R-CNN detector uses the proposed regions for object detection. RPNs generate region proposals with different scales and aspect ratios by using anchor boxes. RPNs and Fast R-CNN share convolutional layers to enhance the running time. Figure 4a shows that both a RPN and Fast R-CNN use the convolutional feature maps. Figure 4b demonstrates that anchor boxes with different scales and aspect ratios. To deploy Faster R-CNN, we used the code base from https://github.com/tensorflow/models/tree/master/research/object_detection.



(a) Faster R-CNN network



(b) Region Propoal Network

Fig. 4. Faster R-CNN [3]

RetinaNet. RetinaNet is proposed by Lin [4]. It tackles the problem that one-stage object detection models fall behind in their accuracy compared to two-stage object detection models. However, one-stage object detection models usually surpass in speed and efficiency to their counterparts. The authors of [4] introduce Focal Loss which is focusing on training hard and incorrectly classified examples by down-weighting easy examples. With the use of Focal Loss, RetinaNet was able to achieve the speed of one-stage detectors without damaging the accuracy.

Figure 5 compares Focal Loss and Cross Entropy Loss. Cross Entropy Loss is a standard loss function commonly used in many of two-stage object detection models such as R-CNN, Fast R-CNN, and Faster R-CNN.

[4] explains cross entropy for binary classification as follows.

$$CE(p,y) = \begin{cases} -log(p) & \text{if } y = 1 \\ -log(1-p) & \text{otherwise} \end{cases} \quad (1)$$

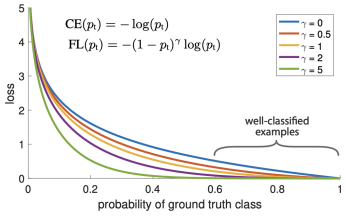


Fig. 5. Focal Loss [4]

From the above, $y \in \{-1, 1\}$ is ground truth class and $p \in [0, 1]$ is the predicted probability of the class. [4] define p_t:

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases} \quad (2)$$

and restate $CE(p, y) = CE(p_t) = -log(p_t)$. The blue curve in Fig. 5 denotes the cross entropy loss.

The focal loss is as defined below and Fig. 5 plots focal loss with different γ values ranging from 0 to 5 [4]:

$$FL(p_t) = -(1 - p_t)^{\gamma} log(p_t) \tag{3}$$

Thus, when $\gamma = 0$, $CE(p_t) = FL(p_t)$. From Fig. 5, the loss is not trivial with easy examples ($p_t \geq 0.5$), and this means that the loss from these easy examples can overwhelm the loss from hard examples when there are much more easy examples than hard examples in the data [4]. [4] found that the results were best when $\gamma = 2$ from their experiments.

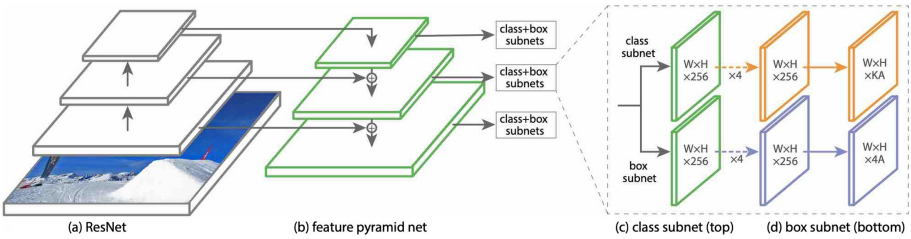


Fig. 6. RetinaNet network architecture [4]

Figure 6 is the architecture of RetinaNet. RetinaNet deploys a Feature Pyramid Network (FPN) on top of ResNet [18] which is a CNN architecture to detect features with deep layers (a). The (b) part depicts FPN constructing a multi-scale feature pyramid and the (c) and (d) parts show the FPN is connected to two sub-networks: one for classification of anchor boxes (Classification Subnet) and the other for regression of anchor boxes (Box Regression Subnet) [4]. To deploy RetinaNet, we used the code base from https://github.com/fizyr/keras-retinanet.

4 Results

4.1 Evaluation Metrics

In object detection, the most common metrics to measure the performance of an object detection model is mean average precision (mAP). mAP is the mean of average precision (AP) of classes which the model try to detect. AP of each class can be calculated using precision-recall curve. Precision-recall curve is plotted with precision and recall values which are calculated using Intersection over Union (IoU) between ground truth bounding boxes supplied from the dataset and predicted bounding boxes from the model. IoU is defined as the following.

$$IoU = \frac{Intersection}{Union} \tag{4}$$

After the object detection model predicts bounding boxes, each predicted bounding box's IoU against its corresponding ground truth bounding box is calculated. If the calculated IoU is greater than the IoU threshold, the predicted bounding box is counted as a true positive (TN). In this study, 0.5 was used for the IoU threshold. However, the predicted bounding box is counted as a false positive (FP) if its calculated IoU is less than the IoU threshold or there is a mismatch with the class between the predicted bounding box and the corresponding ground truth bounding box. Also, if there are more than one predicted bounding box to a ground truth bounding box, the predicted bounding box with the highest IoU with the correct class is counted as a true positive whereas the remaining predicted bounding boxes are counted false positives. The ground truth bounding boxes with no detection are counted as false negatives (FN). For object detection, true negatives (bounding boxes with no object) are not counted because there can be so many possible true negatives in an image. Due to this reason, precision and recall, instead of accuracy, are used to evaluate the performance of a model in object detection. Precision and recall are calculated with TP, FP and FN like the below.

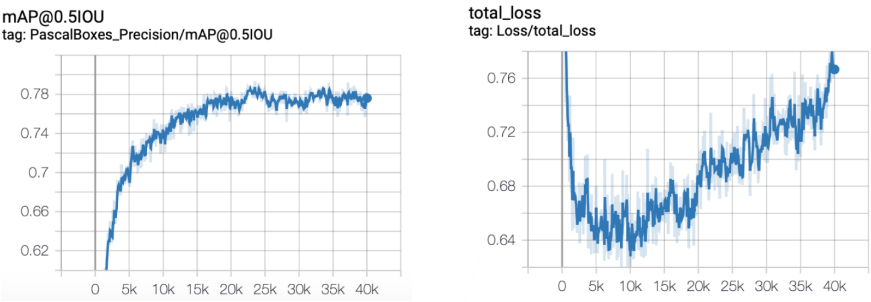
$$Precision = \frac{TP}{TP + FP} = \frac{\text{correct predictions}}{\text{all predictions}} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} = \frac{\text{correct predictions}}{\text{all ground truth objects}} \tag{6}$$

Then, the predicted bounding boxes are sorted according to their confidence value, which is calculated by the object detection model, in descending order (boxes with the highest confidence first). This confidence value is the probability whether the predicted bounding box contains an object of the classes which the object detection model attempts to detect. With each prediction bounding box, the precision and recall is calculated and they are plotted in the precision-recall curve. AP of an object class is calculated by "averaging precision across recall values from 0 to 1" [19]. There are two ways to find AP, namely 11-point interpolation and all point interpolation. We used all point interpolation to find AP. After finding an AP for an object class, mAP is found by averaging APs of object classes. [19] contains more details how to calculate AP and mAP.

4.2 Loss and mAP of Faster R-CNN

Figure 7a plots mAP while training the Faster R-CNN model whereas Fig. 7b plots the loss. The highest mAP for Faster R-CNN was 0.78, which was lower than mAP from RetinaNet. The object detected images (Fig. 8 and Fig. 9) displayed were generated from Faster R-CNN. The left-side images labeled as detected are detection images from Faster R-CNN while the right-side images labeled as ground-truth are ground-truth images. Figure 8 shows images with correctly detected objects while Fig. 9 shows images with some incorrectly detected objects.



(a) mAP for epochs (b) Total Loss for epochs

Fig. 7. Training results for epochs of Faster R-CNN network with ResNet101

Fig. 8. Object detected images with no error generated from Faster R-CNN

4.3 Loss and mAP of RetinaNet

Figure 10a plots mAP while training the RetinaNet model whereas Fig. 10b plots the loss. Figure 11 are precision recall curves and APs of the object classes. The precision recall curves were plotted using a tool available from https://github.com/rafaelpadilla/Object-Detection-Metrics. Also, [20] is the paper about the tool. RetinaNet converges early. At Epoch 3, training mAP reached 0.87 which was the higest mAP. From this results, we found that RetinaNet returns the better results than Faster R-CNN. The object detected images (Fig. 12 and

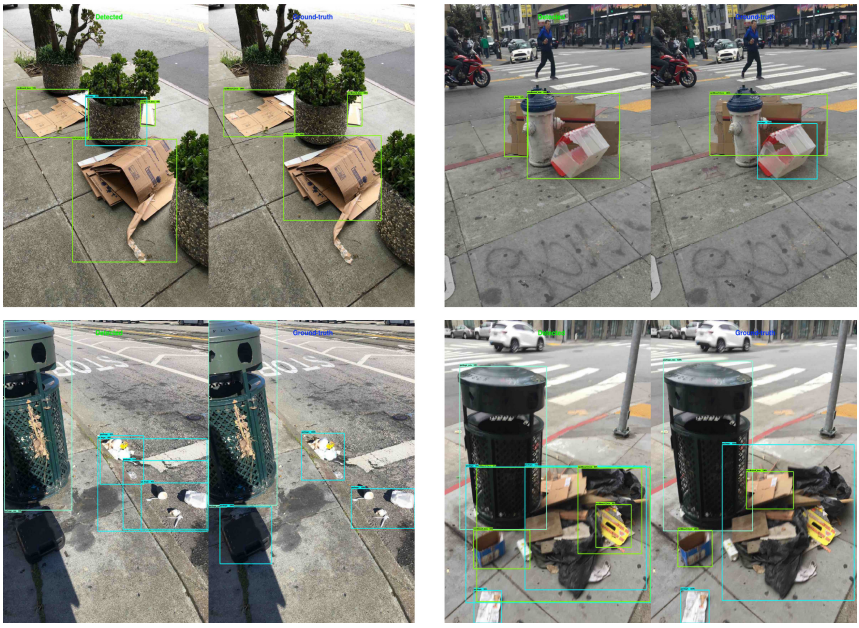
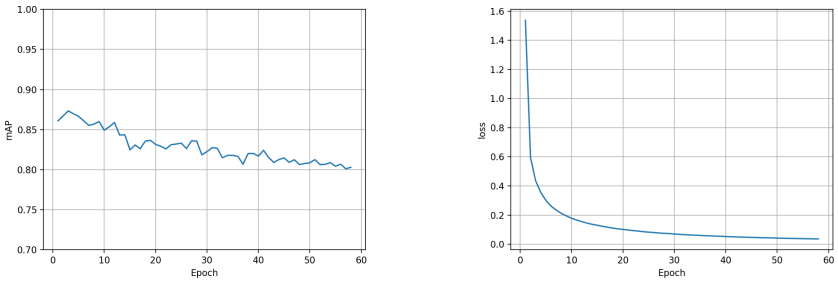


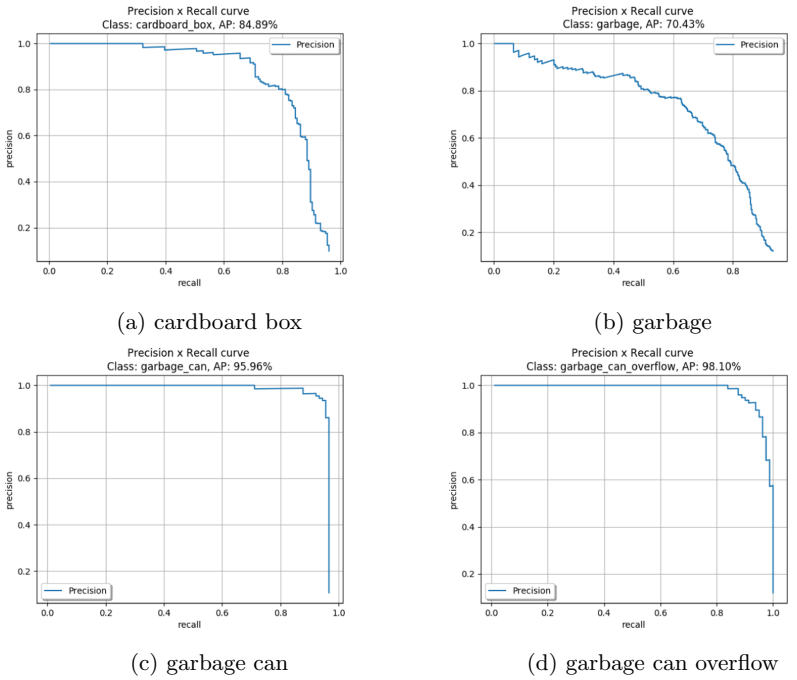
Fig. 9. Object detected images with errors generated from Faster R-CNN



(a) mAP for epochs (b) Total Loss for epochs

Fig. 10. Training metrics for epochs of RetinaNet with ResNet152

Fig. 13) displayed were generated from RetinaNet. The bounding boxes in the images were color-coded according to ground truth (blue), true positive (green), false positive (red) and false negative (yellow). Figure 12 shows images with correctly detected objects while Fig. 13 shows images with some incorrectly detected objects.



(a) cardboard box (b) garbage

(c) garbage can (d) garbage can overflow

Fig. 11. Precision Recall curves

Fig. 12. Object detected images with no error generated from RetinaNet (Color figure online)

Fig. 13. Object detected images with some error generated from RetinaNet (Color figure online)

5 Discussions, Conclusions and Future Work

Citizen science today is becoming extremely useful for a variety of applications that target the greater good. The ubiquity of the Internet, smart-phones and connectivity are natural enablers for citizen-science. While there are services using citizen science to monitor disease outbreaks [21,22], protecting bio-diversity [23,24], pollution monitoring [25,26] and many more, we are not aware of any particular work that focuses on citizen-science and image processing techniques for civic engagement like the ones we are focusing on in this paper. This is the gap, we address in this paper.

Many large cities in the United States employ the 311 service to allow their citizens to report non-emergent issues or to inquire city-related information with various communication channels such as phone, mobile apps, website and Twitter. With these various communication channels, 311 users can not only report an issue but also upload related images. We observed from SF 311 (San Francisco 311) that there are many garbage related images uploaded by its users. With these garbage related images, we implemented object detection with two pre-trained models, namely Faster R-CNN and RetinaNet, to detect four object classes, which are garbage, garbage cans, garbage can overflow and cardboard boxes. RetinaNet outperformed Faster R-CNN with mAP = 0.87 while Faster R-CNN's mAP was 0.78. This object detection on garbage can be used to facilitate automatic garbage detection.

Our future work is to include more classes to detect, and expand our model and database of image to other cities. Looking at GPS locations, will also enable us generate new results and studies related to factors like racial, socio-economic, crime propensity and other important aspects of civic life, as it pertains to quality of civic engagement and how to improve it. But the AI techniques we design in this paper are still robust foundations for such a study. Presenting our studies to policy makers, and getting their feedback using sound HCI designs, and using lessons learned for tangible action is part of our on-going work also.

References

1. Home, SF311: https://sf311.org/home. Accessed 4 May 2020
2. What is 311? https://www.govtech.com/dc/articles/What-is-311.html. Accessed 2 May 2020
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks (2016). arXiv:1506.01497
4. Lin, T., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection (2018). arXiv:1708.02002
5. State of the Art Object Detection - use these top 3 data augmentations and Google Brain's optimal policy for training your architecture: https://medium.com/@lessw/state-of-the-art-object-detection-use-these-top-3-data-augmentations-and-google-brains-optimal-57ac6d8d1de5. Accessed 23 Nov 2020
6. Socrata, Data & Insights: https://www.tylertech.com/products/socrata. Accessed 5 May 2020
7. LabelImg · PyPI: https://pypi.org/project/labelImg/. Accessed 8 May 2020
8. Training Custom Object Detector - TensorFlow Object Detection API tutorial documentation: https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/latest/training.html. Accessed 8 May 2020
9. Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., Qu, R.: A survey of deep learning-based object detection. IEEE Access **7**, 128837–128868 (2019)
10. Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., Pietikäinen, M.: Deep learning for generic object detection: a survey (2018). arXiv:1809.02165
11. Zhao, Z., Zheng, P., Xu, S., Wu, X.: Object detection with deep learning: a review (2019). arXiv:1807.05511
12. Cai, W., Li, J., Xie, Z., Zhao, T., Lu, K.: Street object detection based on faster R-CNN. In: Proceedings of the 37th Chinese Control Conference, pp. 9500–9503. Wohan, China (2018)
13. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: Neural Information Processing Systems (2012)
14. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (2012)
15. Chen, Y., Li, W., Sakaridis, C., Dai, D., Gool, L.: Domain adaptive faster R-CNN for object detection in the wild (2018). arXiv:1803.03243
16. Fan, Q., Brown, L., Smith, J.: A closer look at faster R-CNN for vehicle detection. In: Proceedings of 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden (2016)
17. Girshick, R.: Fast R-CNN. In: IEEE International Conference on Computer Vision (ICCV) (2015)

18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015). arXiv:1512.03385
19. Evaluating Object Detection Models: Guide to Performance Metrics: https://manalelaidouni.github.io/manalelaidouni.github.io/Evaluating-Object-Detection-Models-Guide-to-Performance-Metrics.html. Accessed 29 July 2020
20. Padilla, R., Netto, S., Silva, E.: A survey on performance metrics for object-detection algorithms. In: 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), pp. 237–242 (2020)
21. Shahid, F., Ony, S., Albi, T., Chellappan, S., Vashistha, A., Islam, A.: Learning from tweets: opportunities and challenges to inform policy making during dengue epidemic. In: Proceedings of the ACM on Human-Computer Interaction, vol. 4, CSCW1, Article 65 (2020)
22. Wiggins, A., Wilbanks, J.: The rise of citizen science in health and biomedical research. Am. J. Bioeth. **19**(8), 3–14 (2019). https://doi.org/10.1080/15265161.2019.1619859
23. Tweddle, J., Robinson, L., Pocock, M., Roy, H.: Guide to citizen science: developing, implementing and evaluating citizen science to study biodiversity and the environment in the UK. Natural History Museum and NERC Centre for Ecology & Hydrology for UK-EOF (2012)
24. Nugent, J.: iNaturalist: citizen science for 21st-century naturalists. Sci. Scope **41**(7), 12–14 (2018). National Science Teachers Association
25. Huddart, J., Thompson, M., Woodward, G., Brooks, S.: Citizen science: from detecting pollution to evaluating ecological restoration. In: Wiley Interdisciplinary Reviews Water, vol. 3, issue 3, pp. 287–300. Wiley, Great Britain (2016)
26. Jerrett, M., et al.: Validating novel air pollution sensors to improve exposure estimates for epidemiological analyses and citizen science. Environ. Res. **158**, 286–294 (2017). https://doi.org/10.1016/j.envres.2017.04.023