See the E-Waste! Training Visual Intelligence to See Dense Circuit Boards for Recycling

Ali Jahanian, Quang H. Le, Kamal Youcef-Toumi
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{jahanian,quangle,youcef}@mit.edu

Dzmitry Tsetserukou
Skolkovo Institute of Science and Technology
Moscow, Russia, 143026
D.Tsetserukou@skoltech.ru

Abstract

The state-of-the-art semantic segmentation and object detection deep learning models are taking the leap to generalize and leverage automation, but have yet to be useful in real-world tasks such as those in dense circuit board robotic manipulation. Consider a cellphone circuit board that because of small components and a couple of hundred microns gaps between them challenges any manipulation task. For effective automation and robotics usage in manufacturing, we tackle this problem by building a convolutional neural networks optimized for multi-task learning of instance semantic segmentation and detection while accounting for crisp boundaries of small components inside dense boards. We explore the feature learning mechanism, and add the auxiliary task of boundary detection to encourage the network to learn the objects' geometric properties along with the other objectives. We examine the performance of the networks in the visual tasks (separately and all together), and the extent of generalization on the recycling phone dataset. Our network outperformed the state-of-the-art in the visual tasks while maintaining the high speed of computation. To facilitate this globally concerning topic, we provide a benchmark for E-waste *visual tasks research, and publicize our collected dataset and code, as well as demos on our in-lab robot at* *https://github.com/MIT-MRL/recybot*.

1. Introduction

Recycling *E-Waste* is not only environmentally friendly but also economically rewarding and scientifically a challenging problem. While the speed of designing and producing new electronic devices is creating opportunities for us to innovate and implement new ideas and technologies, the urge for a sustainable advancement is becoming more salient. The rather awakening global statistics [2, 17] about producing millions of metric tons *E-Waste* per year, with only about 20% recycling seems to be a call-out for the AI and robotics community: This percentage, although low, mainly means destroy-then-melt, an easy one-solution-for-all, without much of intelligence. Although melting 1 million phones could potentially recover 16,000kg of copper, 350kg of silver, 34kg of gold and 15kg of Palladium, it not only is causing environmental and health hazards to local communities, but is driven by low-cost rather than higher revenue which ignores the opportunity for more advanced recycling [31]. How can we build intelligent solutions to automate semantic detection and categorization of such objects, to leverage the autonomous robotics manipulation and disassembling tasks?

The problem is scientifically interesting because of the special design of the objects. Consider a cellphone circuit board that because of small components and micron-scale gaps (clearance) between them challenges any manipulation task. Components have a limited pose (and thus limited viewpoints) making it difficult to capture RGB-D or point clouds with an effective precision without using a several-thousand-dollar high-end 3D scanner. Moreover, the amount of data is limited due to the specificity of design from each manufacturer and the way the components are fit together which often causes occlusion. So any visual intelligence solution has to meet several criteria: detection with respect to special sizes (e.g., screws vs batteries), semantic segmentation (i.e., pixel to pixel prediction of the components' labels), instance segmentation (e.g., being able to count two connected cables), and crisp boundaries detection (e.g., finding a clear gap between two components for sending an actuator for prying/pulling out the component). Furthermore, ideally, the learning has to take into account the extent of performing all of these tasks in the vicinity of having diverse designs of components from different brands and manufacturers (e.g., different shapes for a cable or same black color for all the components).

While current deep learning AI models, specifically Convolutional Neural Networks (CNNs or ConvNets), have demonstrated outstanding achievements in training for visual intelligence in each of the above tasks (object detection, semantic instance segmentation, and crisp boundaries detection) with reasonable power of generalization on natural scene images, they are yet to be utilized for our problem. When it becomes to detection of small objects with limited data, ConvNets are more prone to under/over-fitting. ConvNet hierarchy supports for encoding high-level vision and semantics into the higher (deeper) levels of the network, and thus capturing objects and context, but it also loses the spatial details around boundaries of the objects. An interesting observation is that object cores have different probability distributions than their boundaries [13]. So it is natural that weakly correlated boundaries' pixels result in low activations than the highly correlated cores, and thus get washed away during transition between layers and levels (unless we force the network to learn a specific distribution). But how can ConvNets be modified to account for details in our problem?

We tackle the multi-faceted dense circuit problem by building upon state-of-the-art ConvNets and end-to-end (no pre/post-processing) optimizing for multi-task learning of detection, localization, and instance semantic segmentation while accounting for crisp boundaries of small components. Specifically, our method extends the Mask R-CNN [11] by addressing the inherent problems of ConvNets – the trade-off between semantic top levels and detailed bottom levels. First, we explore the proposal feature by pooling feature maps from all levels (top and bottom) of the backbone while benefiting from lateral skip-connections between all levels. The key idea is to regain the higher spatial resolution information of the objects' boundaries from the feature maps at lower-levels of the CNN hierarchy. Second, we add an auxiliary branch for predicting objects' boundaries on each Region of Interest (RoI), in *parallel* with the existing branch for classification, bounding box regression, and masks segmentation. During training, the feature maps produced by the edge branch is cascaded with that of the mask branch to compute the final segmentation mask (Figure 1). This encourages the network to learn the location of boundaries and implicitly captures the geometric properties of phone components along with the others objectives. Thereby, we achieved increased accuracy for the segmentation of our phone dataset by over 3% of standard metric, and more importantly, retained the crisp boundaries for our robotic tasks compared to Mask-RCNN while maintaining the same level of computing speed.

We examine the performance of the networks in the visual tasks (separately and all together), and the extent of generalization, for effective automation and robotics usage in manufacturing. For experiments, we have collected and annotated 533 images of cellphones and their components. We provide a benchmark for *E-waste*, and will release our data and code to facilitate future research.

2. Related Work

We are interested in the end-to-end learning (for creating autonomous systems) of multi-tasking using ConvNets for object detection and localization, semantic segmentations, instance segmentation, and crisp boundaries detection, all together. Learning ConvNets for each of these tasks has been explored separately and benchmarked on public datasets, and yet each is an active topic of research. For example, for semantic segmentation, fully convolutional networks, FCN [22, 25], SegNet [1] and U-Net [29] architectures have been successful. Since these networks do not perform well in details and boundaries, several improvements have been suggested [7, 27, 18].

FPN: Driven by the idea of building high-level semantic feature maps at all scales, recent architectures are based on skip connections from encoder to decoder for using details from different feature maps' level (Laplacian reconstruction in LLR [8], SharpMask [27]). One such architecture is FPN [19] which demonstrates significant improvement as a generic feature extractor in several visual applications. FPN utilizes a pyramidal structure (i.e., multi-level) for encoder and decoder, similar to human visual system in multi-scale visual tasks [4]. FPN further uses an adaptive pooling mechanism for aggregating the feature maps of corresponding encoder/decoder levels together (through lateral skip-connections as well as the inherited layer skip-connections from ResNet [12]) with losses at each level of the decoder. This architecture gives state-of-the-art for any ConNet's backbone. FPN is also used as the backbone in Mask R-CNN [11] which does object detection, localization, and instance semantic segmentation together.

Mask-RCNN is currently state-of-the-art for object detection and instance segmentation, and part of its strength is due to region-based detection mechanism . In fact this network is an evolution of prior work – RCNN [9], Fast RCNN [8], and Faster RCNN [28]. The core idea in this family of networks is to scan over the predefined regions called *anchors*. For each anchor, the Region Proposal Network (RPN) does two different type of predictions: the score of it being foreground, and the bounding box regression adjustment to best fit the object. RPN then chooses a fixed number of anchors with highest score, and apply the regression adjustment to get the final proposal for objects prediction at the network head. This mechanism works better for us (as we compared to SegNet). Note that there are other instance segmentation works [26, 6] that use the region proposals, however, Mask R-CNN learns all the tasks at the head (i.e., for inference).

It is also conceivable to train ConvNets for edge and boundaries detection [32, 30, 5, 33, 10], e.g., HED [32] uses loss on edge maps at different levels of a ConvNet to force the network for finding boundaries in images. While finding edges and object cores could be two different compet-

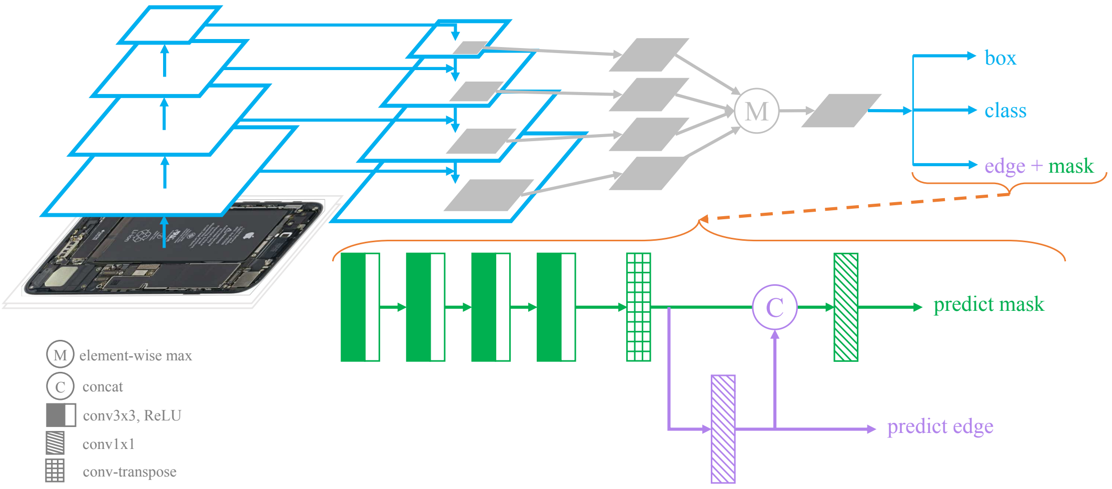

Figure 1. Network architecture. The blue path illustrates the baseline model with box, class, and mask at the head. Note that the green mask path is different from the implementation in the baseline. We modify the pooling mechanism from the backbone and illustrate it via the gray path. When we add only the green mask, we call it *Mask R-CNN + FPN + Adaptive Pooling*. When we add the purple edge path as well, we call it *Mask R-CNN + FPN + Adaptive Pooling + Edge Detection*. Note for avoiding clutter, we did not draw depth of the feature maps. Also, the icons in left-bottom corner apply to all colors.

ing tasks, there are works that explore doing both tasks as a multi-task learning [15, 14, 23, 3]. The main idea is a combination of a loss function that accounts for both tasks and one or several networks that learn features for each task. We learn the edges on the same network of other tasks, and in our implementation, we concatenate (using 1×1 convolution) edge learned feature maps to the ones for segmentation mask. That is, we want the edge detection to be an auxiliary task that forces the network to be aware of boundaries.

3. Method

Our goal is to semantically detect and localize components with clean boundaries. For this, we want to devise a method that learns multi-tasks in an end-to-end learning fashion. Our method is borrowed from the concept of region-based detection for instance-segmentation. For modeling we take Mask R-CNN as the baseline and build upon it. We first discuss the objective and the multi-tasking and then describe the architecture in details.

3.1. Objective

We want to optimize our learning with respect to several tasks. Technically, besides the three inference terms, the bounding box, class, and mask, we also want a geometric term which incorporates the boundary information of the segmentation mask into a single loss function.

In the baseline Mask R-CNN model, three tasks are done at the inference. Formally, the model optimizes the following total loss function:

$$L = L_{bbox} + L_{class} + L_{mask}. \tag{1}$$

Here, L_{bbox}, L_{class}, and L_{mask} correspond to the L_2 regression loss for the bounding box of each component (localization), class-agnostic presence of a component (detection), and per-pixel label (semantic segmentation), respectively. In our case, we also want to emphasize on the boundaries of the components to find clear gaps between components for manipulation tasks, e.g., prying. Therefore, we explicitly include the semantic edge loss to the total loss as follows:

$$L = L_{bbox} + L_{class} + L_{mask} + \lambda L_{edge}. \tag{2}$$

Here, L_{edge} corresponds to the loss between the edge mask (per-pixel ground truth of boundaries of each component) and the predicted edge mask. To regularize the weight of the edge loss in the total loss, we use λ coefficient, and treat it as a hyper-parameter in our experiments.

3.2. Network Architecture

Our goal is to design end-to-end ConvNets for learning with the objectives formulated in Equation (1) and Equation (2). We compare the results of these networks with the baseline in Sec. 4.

Figure 1 illustrates the network architecture with different colors to highlight different parts and designs. The blue path shows the schematic structure of the base model: it starts with an FPN backbone (with fully enabled lateral connections) [19], and ends with the three branches for bounding box, class, and mask predictions to correspond to the objective defined in Equation (1). For the FPN backbone, we used *ResNet50* architecture with initialized weights from ImageNet. Because we have a different design for the mask branch, we replaced the original one with “edge + mask” in purple and green (instead of blue). Specifically, in our design, there are two main different parts in the adaptation of the base model, as follows:

3.2.1 Feature Pooling

This part refers to the gray part in Fig. 1, and corresponds to the different usage of RoIAlign in the base model. That is, we use all levels of feature maps for reconstruction of a mask. The intuition behind this idea is to use information of top levels for context and low levels for details (boundaries in our case). This idea is also explored in [21] and called “Adaptive Feature Pooling.” We stick to this term in our work; and further, we note that they pool these features from an extra (shallow multi-scale) encoder following the FPN backbone. We pool (max-element-wise) the proposals computed by RoIAlign from the decoder of the FPN backbone.

3.2.2 Network Head

We now explain the head branch for mask prediction –the green and purple paths (in Fig. 1). We denote $m \times m$ for size of the mask, and K for number of component classes. After the adaptive feature pooling, we send the RoIAligned feature maps (proposals) with size of $\frac{m}{2} \times \frac{m}{2} \times depth$ to the green path to apply four convolutions followed by ReLU and then a convolution-transpose to upsample to the same size of ground truth mask, $m \times m$. This design concludes our green path “Mask-RCNN + FPN + Adaptive Pooling” network (in Sec. 4) for Equation (1). To implement Equation (2), we add the purple path: We take the convolution-transposed feature maps and first learn the edges by applying a 1×1 convolution and a sigmoid which results in an $m \times m \times K$ prediction. We also take the edge feature maps (before prediction) and concatenate it with the mask feature maps to support the mask prediction with the edge information. This design concludes our purple and green paths together which we refer to as ”Mask-RCNN + FPN+Adaptive Pooling + Edge Detection” network (in Sec. 4), for Equation (2).

4. Experiments

We ran experiments using the architecture in Fig. 1 on a dataset of cellphones. Here, we discuss the details on the dataset, implementation, and evaluation metrics. Furthermore, we discuss an experiment to examine the extent of generalization on the framework for our data.

4.1. Dataset

We have collected a dataset of cellphone images with their annotations. This dataset includes 10 cellphones we have disassembled in the lab. We have taken images of each layer of a phone as well as its individual components. So far, we have collected data for *Apple iPhone 3GS*, *iPhone 4*, *iPhone 4S*, *iPhone 6*, *Samsung GT-i8268 Galaxy*, *Samsung S4 Active*, *Samsung Galaxy S6* and *S6 Edge*, *Samsung S8 Plus*, *Pixel 2 XL*, *Xiomi Note*, *HTC One*, *Huawei Mate 8, 9, 10*, and *P8 Lite*. Moreover, we have collected images from online search engines. We created a taxonomy of components and parts, of which we selected 11 classes. We annotated all the images according to these classes. So far, we have collected over 533 images.

We first split data to 90% for training and 10% for validation, and then apply augmentation on the training set. We add random noise, Gaussian blur, sharpness, and random change in lightness, as well as constant normalization. This results in over 4,000 training images. For validation, we take at least one image from each cellphone model such that it is not shown to the model during training. However, we note that because each cellphone has several images for different layers, there might be a chance that the model sees some components (e.g., camera) stayed in several layers. This effect is inevitable and mainly happens in the real-world scenarios. Nonetheless, to examine the generalization capacity of the modeling, we test the model against the two held-out cellphone models.

4.2. Implementation Details

We have tried different sizes of RoIAlign, mask size ($m \times m$ in Sec. 3.2.2), and coefficient λ in Equation (2). Although, it would be intuitive to take RoIAligns relative to the size of each level in FPN, we found that one size of 14×14 gives us the best results. This counter-intuitive effect could be due to the fact that RoIAlign is already an approximation of feature maps and a large size means having more interpolated activations than the actual activations. We experienced that mask size of $m \times m = 28 \times 28$ (same as in the baseline model) is much better than larger sizes. Finally, we found that the hyper-parameter $\lambda = 1.2$ gives us the best results for component boundaries detection. Figure 2 illustrates these results for a few number of cellphones in the dataset.

4.3. Evaluation Metrics

The standard metrics for pixel to pixel segmentation are mainly the COCO [20] average precision (AP) metrics: AP is average precision of multiple IoU’s (Intersection of prediction and ground truth over Union of prediction and ground truth) values. AP_{50} and AP_{75} are average precision at IoU threshold of 0.5 and 0.75. AP_S, AP_M, AP_L are the average precision for objects with different area scales. Area is measured as the number of pixels in the segmentation mask. Small object: $area < 32^2$, Medium object: $32^2 < area < 96^2$, Large object: $area > 96^2$. In Table

Model	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Mean F-measure
Mask-RCNN + FPN	54.9	70.6	62.5	29.9	46.4	77.2	0.329
Mask-RCNN + FPN + Adaptive Pooling	56.2	72.4	63.2	33.6	**51.8**	77.5	0.349
Mask-RCNN + FPN + Adaptive Pooling + Edge Detection	**58.7**	**73.8**	**65.0**	**35.3**	49.4	**80.1**	**0.363**

Table 1. Mean of mask and boundaries for the three models. AP's are COCO average precisions for segmentation mask. Mean F-measure is the arithmetic mean of F-measure (harmonic mean) of all classes for boundaries.

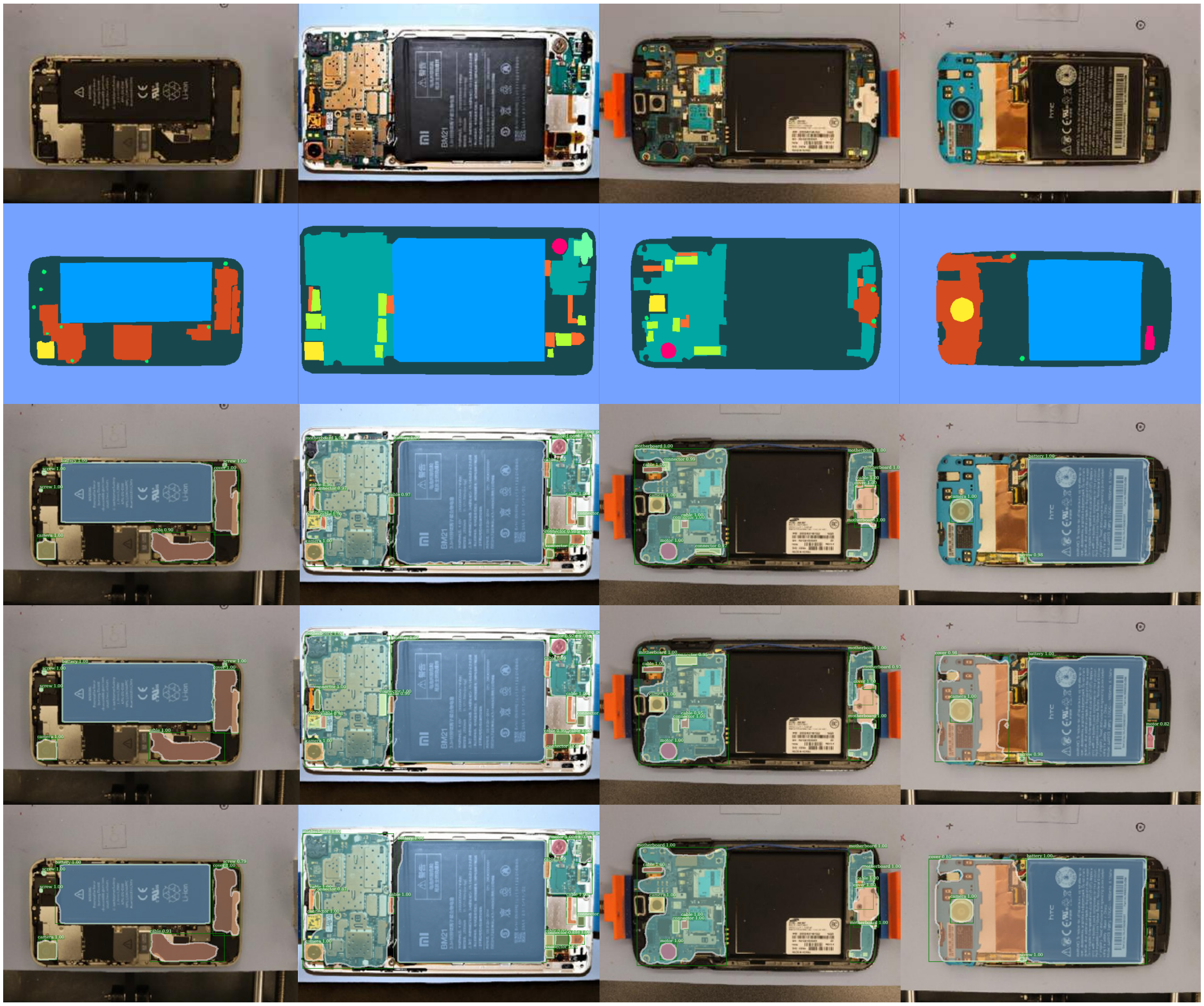

Figure 2. Results of our method. From top, rows are: Input image, ground truth, Mask R-CNN + FPN, Mask R-CNN + FPN + Adaptive Pooling, and Mask R-CNN + FPN + Adaptive Pooling + Edge Detection. Note that the last two rows qualitatively have more accurate (respect to the baseline) boundaries detection for each object, and the last row has even better boundaries detection. Also note that we picked images where we have several objects in the dense board, otherwise doing the multi-tasks on an image of a single component is much easier and less interesting to us.

1, we report these numbers for the three models. Here, we observe that the latest model has highest accuracy almost in all AP metrics, importantly for AP_S.

For evaluating our boundary detection, we compute the F-measure based on the precision-recall curve in [24]. Here precision is number of overlap (between the ground truth and the predicted pixels) over the number of predicted pixels. Intuitively, it means how many pixels the machine correctly captured. On the other hand, recall means number of overlap over the number of ground truth pixels, accounting for how much the machine missed to capture the ground truth. Note that our goal was not to capture the human bias

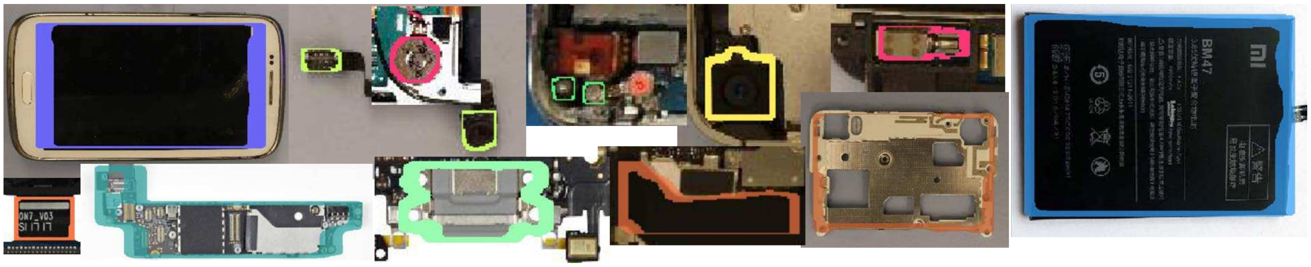

Figure 3. Visualization of boundaries (edges) predicted by Mask R-CNN + Adaptive Pooling + Edge Detection model. For the interest of space, we only show patches containing the object of interest. For each class, we use its color legend and draw the boundaries. Note that the thicknesses of the boundaries are only for visualization, and we did not normalize it across all the classes.

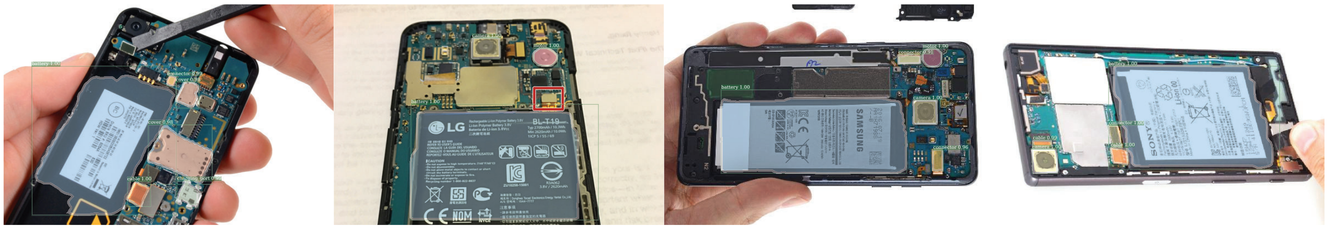

Figure 4. Examining generalization capability of the model: Testing unseen cellphone models from online (ifixit.com [16]) images. For testing, we use Mask R-CNN + Adaptive Pooling + Edge Detection. These cellphone models are, from left: *BlackBerry Z10*, *Nexus 5X*, *Samsung Note 8*, and *Sony Xperia*. The detection of components are reasonable, however, in the real-world scenarios and for manufacturing purposes one would need to train the model on an example of any new brand model before using the ConvNet model for prediction to get better results.

in annotating what a boundary is. Furthermore, as noted by Isola et al. [13], for boundary detection, we not only have to find the "correct" pixels, but also need to find the "crispness." The latter term refers to the set of pixels that are correctly detected and reside on the boundaries, as opposed to being closer to the core of the object. While it is a common practice for boundary detection metrics to take about 4 pixels to account for the intersection of ground truth and prediction, we only take 2 pixels in our report.

Finally, to account for perceptual quality that might not be completely captured by the standard measures, we provide results of the three models in Fig. 2. It is important to note that our ground truth are not quite accurate, due to inherent human bias (which in turn could affect the detection), yet reasonable for our attempt. For qualitative examination of how the edge prediction behaves, we show boundaries predicted for each class in few examples in Fig. 3.

4.4. Generalization

To explore how much the model can generalize the task, we performed the following study: We trained on our dataset, and then tested the model on images of cellphone models it has not seen. These images are randomly selected from the Web (ifixit.com [16]), without prior annotations. Here, we show qualitative results in Fig. 4. We note that our model can produce reasonable predictions for both mask and edge on unseen phone models. Furthermore, this proves that the extra edge branch does help our model to learn the common phone components geometry (e.g., battery is rectangular). Nonetheless, prior work on model generalization indicates that obtaining more (in terms of diversity but not necessarily quantity) data could help for better generalization. We note that our current focus is multi-tasking on small objects and boundaries, and a formal investigation of generalization is an important direction of future work.

5. Conclusion

In this paper, we addressed the problem of visual intelligence systems for autonomous robotics manipulation and disassembling tasks, as such intelligent operations are essential for effective and low-cost *E-Waste* recycling. We explored several designs of ConvNets for end-to-end multitasking of detection, localization, instance semantic segmentation, and boundaries detection. We examined these designs on a dataset of cellphones we collected. We showed how our methods can improve segmentation accuracy while retaining the crisp boundary predictions suitable for dense boards. Our proposed method outperformed the state-of-the-art in both mask segmentation and edge detection, showing the effectiveness of our work. To facilitate research in this direction, we provide a benchmark along with data and code to the community. Finally, we invite the community to participate in this challenge.

ACKNOWLEDGMENT

The authors thank the Skoltech-MIT program for their support. The first author, AJ, would like to thank Phillip Isola and Wenzhen Yuan for the discussions. We would like to thank students Wilhelmena Figueiredo, Andrew Luo, Yingnan Cui, You Wu, Moin Nadeem, Tony Tohme, John Lin, Yuria Utsumi, Victor Cheng, and Michel Finn.

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. 2

[2] Cornelis P Balde, Vanessa Forti, Vanessa Gray, Ruediger Kuehr, and Paul Stegmann. *The global e-waste monitor 2017: Quantities, flows and resources*. United Nations University, International Telecommunication Union, and International Solid Waste Association, 2017. 1

[3] Benjamin Bischke, Patrick Helber, Joachim Folz, Damian Borth, and Andreas Dengel. Multi-task learning for segmentation of building footprints with deep neural networks. *arXiv preprint arXiv:1709.05932*, 2017. 3

[4] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in Computer Vision*, pages 671–679. Elsevier, 1987. 2

[5] Liang-Chieh Chen, Jonathan T Barron, George Papandreou, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with task-specific edge detection using cnns and a discriminatively trained domain transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4545–4554, 2016. 2

[6] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. In *European Conference on Computer Vision*, pages 534–549. Springer, 2016. 2

[7] Golnaz Ghiasi and Charless C Fowlkes. Laplacian pyramid reconstruction and refinement for semantic segmentation. In *European Conference on Computer Vision*, pages 519–534. Springer, 2016. 2

[8] Ross Girshick. Fast r-cnn. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 2

[9] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. 2

[10] Zeeshan Hayder, Xuming He, and Mathieu Salzmann. Boundary-aware instance segmentation. In *30Th Ieee Conference On Computer Vision And Pattern Recognition (Cvpr 2017)*, number CONF. Ieee, 2017. 2

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[13] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H Adelson. Crisp boundary detection using pointwise mutual information. In *European Conference on Computer Vision*, pages 799–814. Springer, 2014. 2, 6

[14] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bogdan Savchynskyy, and Carsten Rother. Instancecut: from edges to instances with multicut. In *CVPR*, volume 3, page 9, 2017. 3

[15] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386*, 2015. 3

[16] Kyle Wiens. iFixit, 2017. `http://www.ifixit.com/`, Last accessed on 2018-09-15. 6

[17] Brook Larmer. E-waste offers an economic opportunity as well as toxicity. 1

[18] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian D Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Cvpr*, volume 1, page 5, 2017. 2

[19] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 2, 3

[20] 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 *European conference on computer vision*, pages 740–755. Springer, 2014. 4

[21] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8759–8768, 2018. 4

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[23] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional oriented boundaries: From image segmentation to high-level tasks. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):819–833, 2018. 3

[24] David R Martin, Charless C Fowlkes, and Jitendra Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE transactions on pattern analysis and machine intelligence*, 26(5):530–549, 2004. 5

[25] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 2

[26] Pedro O Pinheiro, Ronan Collobert, and Piotr Dollár. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998, 2015. 2

[27] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016. 2

[28] Shaoqing Ren, 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*, pages 91–99, 2015. 2

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2

[30] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2015. 2

[31] United States Environmental Protection Agency. The secret life of a smart phone. 1

[32] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 2

[33] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. Casenet: Deep category-aware semantic edge detection. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, pages 21–26, 2017. 2