Efficient Network For Applying Trash Segmentation into Real Life

Minh-Khoi Pham, Dang-Khoa Nguyen, Hoang-Lan Nguyen
University of Science, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
{ 18120043, 18120040, 18120051 } @student.hcmus.edu.vn

***Abstract*—Environmental littering has always been a difficult and complex worldwide problem, thus more solutions are demanded. Nowadays, with the growth of artificial intelligence, many unsolvable tasks can be effectively analyzed and tackled. Thereby, we hope that our research will provide an in-depth overview and useful tools for dealing with the mentioned problem by using a deep neural network to detect and classify trash in the environment. We see this as an instance segmentation problem, which is separating trash from the scene, therefore, apply the state-of-the-art model: YOLACT. We also propose modified versions which we call Efficient YOLACT. As a result, the model performs reasonably well on the datasets and may give a desirable results when applying to the real-life problem.**

I. Introduction

Due to the increase in population and economic development, there are more and more trashes is produced [8]. There are six types of waste [4], each one different shape, the danger level, and process way. To choose a proper treatment for each one, separating from the scene and segment it from each other is needed.

Manually, trash is often classified by people at the landfill. These manual labor jobs are usually dull, repetitive, in some cases, waste can pose a threat to human health [1], which has been replaced by high-technology machines. Such as automatically separated in materials recovery facilities or mechanical biological treatment systems. However, these machines still very big and expensive, cost per ton ranges from below €50 to above €500 for the whole facility [12]. In this paper, we will perform a better solution by using the power of computer vision. This often results in much higher performance than manually-done by human.

Object detection and segmentation are always hot topics in computer vision, where waste objects are much more challenging and less attractive than most other ones [13]. Due to a large number of small objects, it is difficult to even for most humans to accurately delineate waste object boundaries without know exactly what these are. For the human vision system, attention can either be shifted to cover a wide area of the visual field or narrowed to a tiny region as when we scrutinize a small area for details. Trying to achieve that, we think YOLACT could be applied to trash instance segmentation by using the TACO dataset. While trying to solve this problem, we recognized the YOLACT model is using the old model - ResNet. Then we try the EfficientNet [11] instead, we replace EfficientNet at both feature backbone and feature pyramid. Surprisingly, the result of the experiment shows that the model acts well on most of the images. Therefore, we hope our work is helpful in many fields especially in robotics.

Fig. 1. Some samples from TACO

The rest of the paper is organized as follows: Our approach is illustrated in III, IV illustrates the results, . In V, conclusions are presented.

II. Related Work

While many real-time object detection and semantic segmentation methods exist, few researches focused on instance segmentation, especially real-time instance segmentation. We decide to solve the trash problem by using the instance segmentation method, putting the strategy of YOLACT [2] and EfficientNet [11] together.

A. Instance segmentation

Most of the instance segmentation methods in common use today are built on two-stage detectors. Fast R-CNN [5] and Faster R-CNN [10] are typical two-stage detectors. Mask R-CNN [6] is representative of a two-stage detector based approach which is used a lot by the community nowadays because of high accuracy in localization and object recognition, but it also has disadvantages:

- Two-stage architecture makes them unable to obtain real-time speed even when decreasing image size.

- ROI pool steps (ROI-align): give the ROIs to model to predict the mask, the sequential processing flow is difficult to parallelize the computation to speed up.

Whereas one-stage detectors like YOLO [9], RetinaNet [7] achieve higher inference speed, they propose predicted boxes from input images directly without the region proposal step so that they are time efficient and can be used for real-time devices.

B. PrototypeNet

PrototypeNet is a Fully Convolutional Network (FCN) which is built on features Full Pyramid Network (FPN). Its output return Prototype Masks which have double upsample size. Prototype Mask is considered to be basic elements, when combined with different ratios it will produce the corresponding mask for each object.

C. YOLACT

YOLACT is built on the backbone of Retina: ResNet and FPN (Feature Pyramid Network). Then the Instance Segment split into two parallel branches, simple and separate the prototype net branch and the prediction head branch. This division into 2 branches optimizes and parallelizes the computation, helps YOLACT achieve real-time speed, 3 to 5 times faster than Mask R-CNN.

D. EfficientNet

The original Yolact uses ResNet and FPN to be the backbone of their model. We replace ResNet with EfficientNet as EfficientNet is a baseline network which achieves better accuracy and efficiency than other's networks. EfficientNet-B7 achieves state-of-the-art on ImageNet Accuracy (84,4% top-1 and 97.1% top-5) and its model is 8.4x smaller than GPipe. One thing we see that makes EfficientNet better than ResNet is because of its compound model scaling. It is a better way to scale up CNN's than other conventional practices.

III. Method

A. Overview

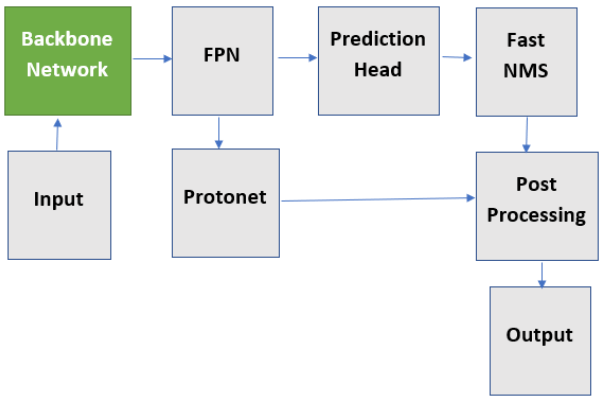

Fig. 2. Efficient YOLACT Architecture

Figure 2 show the our model pipeline. Firstly, the input image is fed to the backbone to extract feature maps at different levels. These maps then are forwarded to two different branches, a prediction head which gives out bounding boxes and mask coefficients, and a Protonet, which is implemented as an FCN [3], predicts a set of prototype masks. Finally, we combine the two outputs by a sigmoid function (same as the one in [3]) to get final object instance masks.

B. EfficientNet backbone

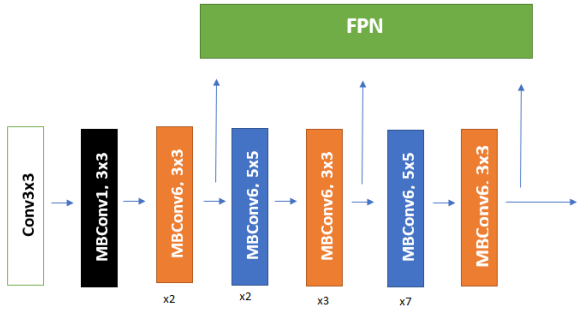

Fig. 3. EfficientNet Structure

We use the original Efficient Net which was proposed by Tan et al [11]. We extract layers at 3 positions, which are outputs from the MBConv6, 3x3 blocks (orange blocks in the figure) and forward to FPN. We choose these layers in order to get multiple features in different sizes, for detecting objects at different scales.

Backbone	#Parameters (Millions)
ResNet50-FPN	33
ResNet101-FPN	52
EfficientNetb0–FPN (ours)	**12**
EfficientNetb6–FPN (ours)	**47**

TABLE I
NUMBER OF PARAMETERS IN THE BACKBONE

Table I shows that our modified backbone has less parameters than the original. This benefits for mobile device due to small-size model's weights.

IV. Experiments

A. Evaluate on COCO

First, we pretrain our model on COCO dataset for evaluation. We modify the model by changing the the original ResNet in the backbone to EfficientNet-B0. We select feature maps at position 4th, 9th and 16nd (figure 3) which are extracted from the backbone to be put into FPN. We keep all other settings from original paper [3] such as loss functions, SGD optimizer, except changing only the learning rates. We start learning rate at 0.001 then divide by 10 at step 200000, 250000, 275000 and train for 300000 iterations. Both models are trained on

COCO2017-train dataset using Tesla T4 with batch size of 4 for B2 version.

For data augmentation, during training, after resizing the input to 550×550, we randomly apply image cropping, mirroring, horizontal flipping, rotating at 90 degrees, changing image's hue, vibrance and contrast.

We test our modified model on COCO2017-val and compare the results with the original. The mAP scores are evaluated using Tesla T4, while the inference time is calculated when running on GTX 1060.

As can be seen in table II, although the accuracy of the original model are higher (1.23 times) than our proposed model, our model's number of computation and iterations are much lower (3.375 times) while achieving little higher speed.

Backbone	#Epochs	AP_{50}	AP_{70}	FPS
ResNet50	54	52.72	40.5	6.62
ResNet101	54	53.17	41.42	5.52
EfficientNetb0 (ours)	**16**	**42.70**	**30.98**	**7.2 – 7.5**
EfficientNetb6 (ours)	**12**	**42.97**	**31.29**	**5.25 – 5.3**

TABLE II
MAP ON COCO2017-VAL

B. Evaluate on TACO

TACO is the new public dataset consist of 60 types of trash which belong to 28 super categories. All samples from the dataset are high resolution images and are annotated for object detection and object segmentation. Furthermore, labels also contain tag for the scene of the background. There are two main challenging problem about the dataset: its tiny size (1500 images with 4784 annotations) and class imbalance (shown in figure 4)

For comparison, we train both base YOLACT model and our Efficient-YOLACT on the TACO dataset for segmenting trash in the scene. In figure 5, it can be seen that the result improves from 16.33 mAP to 20.36 mAP when evaluate using our new model. We conclude that our model can converge faster than the baseline YOLACT model and performs much better when detecting smaller objects.

Backbone	#Iterations	AP_{50}	AP_{70}	AP_{90}
ResNet50	100,000	16.33	13.31	5.6
EfficientNetb2 (ours)	**100,000**	**20.36**	**17.4**	**7.4**

TABLE III
MAP ON TACO-VAL

V. CONCLUSION

Despite not improving much than the original, our model gives an in-depth overview about the TACO dataset, so that help understanding what are the strengths and difficulties when investigate in the one of the most crucial real-life problems. We hope that our approach can provide useful tools for environments observation and can make the automation in environmental topics become more practical.

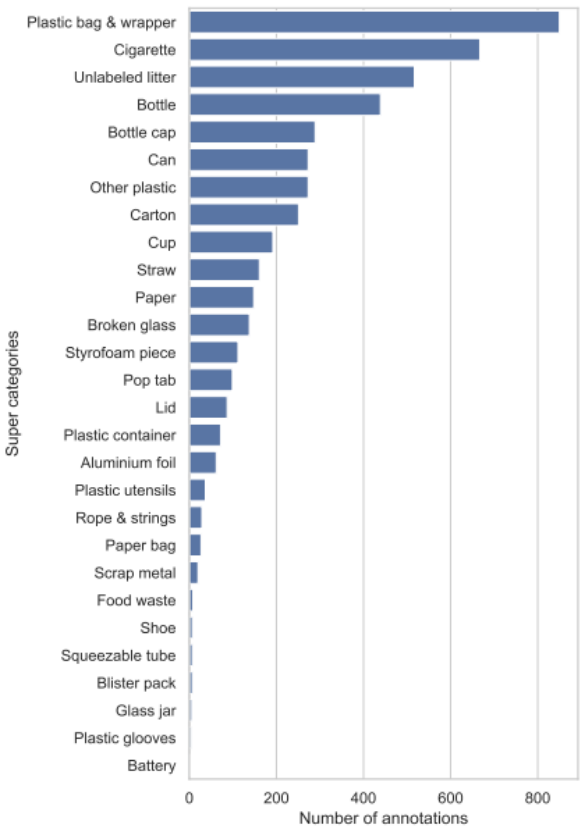

Fig. 4. TACO Distributions

Fig. 5. TACO results comparison

REFERENCES

[1] Editorial board/aims and scope. *Waste Management*, 34(3):IFC, 2014.

[2] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. *CoRR*, abs/1904.02689, 2019.

[3] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. 2019.

[4] Perinaz Bhada-Tata Daniel Hoornweg. *What a waste: a global review of solid waste management*, volume 15. World Bank, Washington, DC, 2012.

[5] Ross Girshick. Fast r-cnn. *CoRR*, abs/1504.08083, 2015.

[6] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2017. cite arxiv:1703.06870Comment: open source; appendix on more results.

[7] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[8] S. C. Wirasinghe Sumith Pilapiiya Nilanthi J. G. J. Bandara, J. Patrick A. Hettiaratchi. Relation of waste generation and composition to socio-economic factors: a case study. *Environmental Monitoring and Assessment*, 135(1–3):31–39, 2007.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2015. cite arxiv:1506.02640.

[10] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015.

[11] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. 2019. cite arxiv:1905.11946Comment: Published in ICML 2019.

[12] Konstantinia Tsilemou and Demetrios Panagiotakopoulos. Economic assessment of mechanical-biological treatment facilities. 39:55–63, 01 2007.

[13] Tao Wang, Yuanzheng Cai, Lingyu Liang, and Dongyi Ye. A multi-level approach to waste object segmentation. *Sensors*, 20:3816, 07 2020.