Leboh 2: An Android Application for Solid Waste Detection

Teny Handhayani, Ageng Hadi Pawening, Janson Hendryli

***Abstract*—Garbage or waste will be a serious problem if they are not managed properly. Sorting garbage is necessary before putting them in the bins. This paper develops an Android application called *Lebooh 2* for garbage detection using EfficientDet-Lite, a Convolutional Neural Network's architecture for object detection. *Lebooh 2* is designed for Indonesian speakers. This paper applies two models: Model 1 and Model 2. Model 1 is trained to recognize 11 types of garbage, i.e., biological, battery, cardboard, metal, glass, plastic, paper, clothes, shoes, electronics, and trash. Model 2 is developed to detect whether the garbage is recyclable or not-recyclable. The experiment using EfficientDet-Lite 3 obtains the mean average precision are around 75% - 77%. User testing involves participants as testers and they collect 400 screenshots of Leboh 2 for solid waste detection. The user testing produces an accuracy of around 78% and 82% for Model 1 and Model 2, respectively. A simple survey of 172 respondents reveals that 22.7% of them have not yet learned how to sort the garbage.**

***Index Terms*—EfficientDet-Lite, Android, garbage, object-detection, waste.**

I. Introduction

MUNICIPAL solid waste (MSW) will be a serious problem if they are not managed properly. Everybody must be aware to participate in waste management, e.g., sorting the waste and putting it in the proper bin. An example of sorting waste is separating the garbage based on its material, i.e., plastic, metal, or glass. The waste also can be grouped into recycling-waste and not recycling-waste. Figure 1 shows examples of well-organized bins from various countries. In some places, sorting waste (garbage) is not well done due to the lack of information on the waste management system and the availability of garbage bins.

Previous studies implement various machine learning and deep learning methods have been done to develop models for image-based garbage classifications [1] [2] [3] [4] [5] [6] [7] [8] [9] [10]. Recently, the most popular method for image classification is Convolutional Neural Network (ConvNet). The advance of ConvNet is this model allows processing images in the raw form [11]. The development of ConvNets is not only used for image and video classification but also for object detection: R-CNN [12], SPP-net [13], Faster R-CNN [14], R-FCN [15], SSD [16], and YOLO [17]. The related works for garbage detection can be read in some publications [18] [19] [20] [21] [22] [23] [24].

Manuscript received May 13, 2023; revised September 26, 2023. This work was supported by a research grant from LPPM Universitas Tarumanagara No 0952-Int-KLPPM/UNTAR/VII/2022.

Teny Handhayani is a lecturer at Computer Science Department, Faculty of Information Technology, Universitas Tarumanagara, Jakarta, Indonesia, 11440 e-mail: tenyh@fti.untar.ac.id.

Ageng Hadi Pawening is a data analyst at Kiriminaja, Yogyakarta, Indonesia, e-mail: agenghadipawening@gmail.com.

Janson Hendryli is a lecturer at Computer Science Department, Faculty of Information Technology, Universitas Tarumanagara, Jakarta, Indonesia, 11440 e-mail: jansonh@fti.untar.ac.id.

The goal of developing a mobile application for garbage detection is to help people understand the basic knowledge of sorting waste. The previous version of *Leboh*, an Android application for waste classification, is only available to classify a single object of garbage and it does not have a feature for locating the specific object in an image [25]. Suppose in an image, there are several objects, e.g., orange, tissue, paper, a glass bottle, and a glove, this application outputs 'biological'. It might produce confusing and misleading information for the users. This problem encourages the authors to improve the performance of *Leboh* using the object detection method.

The research question is how to detect multiple garbage types in an image. The authors propose to use EfficientDet-Lite for garbage detection. The application is called *Leboh 2* and it is expected to help people understand the basic knowledge of sorting waste. This paper develops two models for garbage detection. The first model is available to detect the types of garbage (electronic, glass, biological, battery, cardboard, metal, paper, plastic, clothes, shoes, and trash) and the second model is able to detect the categories of garbage (recyclable and not-recyclable). The contribution of this paper is models and a prototype Android application for garbage detection that is designed friendly for Indonesian speakers. This paper also reports a survey of 172 participants via social media about people's opinions on garbage management.

II. Research Method

A. *EfficientDet*

ConvNets are compatible with data that has the form of multiple arrays. The architecture of the ConvNet is arranged as a series of stages [11]. In the early stages, it has two kinds of layers: convolutional layers and pooling layers. Feature maps consist of units in a convolutional layer. A filter bank is a set of weights that connect each unit to local patches in the feature maps. A non-linearity function, e.g., ReLU, is used to pass the summation of local weighted. In a feature map, all units share the same filter bank. Different filter banks are used by different feature maps in a layer. The stack consists of two or three stages convolution, non-linearity, pooling, more convolutional, and fully connected layers. ConvNets allow the images to proceed in their raw form.

EfficientDet is a ConvNet architecture for object detection. Figure 2 shows EfficientDet architecture [26]. EfficientDet architecture implements EfficientNet, a bi-directional feature pyramid network (BiFPN), and a shared class/box prediction network. EfficientNet is a ConvNet architecture implementing a compound scaling method and maintaining model efficiency [27]. In EfficientDet architecture, EfficientNet and BiFPN are used as the backbone network and the feature network, respectively. A weighted BiFPN is a method for

Fig. 1. Examples of well-organized bins from various countries.

multi-scale feature fusion. BiFPN takes level 3-7 features from the backbone network and repeatedly applies top-down and bottom-up bidirectional feature fusion. P_i refers to a feature level, where $i = 1, 2, ..., 7$. The fused features are used to feed a class and box network to produce object class and bounding box predictions respectively. A compound scaling method for object detection implements a simple compound coefficient ϕ to jointly scale up all dimensions of the backbone network, BiFPN network, class/box network, and resolution. Grid search for all dimensions is prohibitively expensive because the object detector has more scaling dimensions than the image classification model. EfficientDet architecture implements a heuristic-based scaling approach but still follows the main idea of jointly scaling up all dimensions.

B. Evaluation methods

In binary case classification, true positive (TP) and false positive (FP) refer to the number of correctly and incorrectly predicted positives. The true negative (TN) and false negative (FN) refer to the number of correctly and incorrectly predicted negatives. The standard precision and recall are defined by equations (1) and (2) respectively.

$$Precision = \frac{\sum_{i=1}^{j} TP}{\sum_{i=1}^{j} TP + FP} \quad (1)$$

$$Recall = \frac{\sum_{i=1}^{j} TP_i}{\sum_{i=1}^{j} TP_i + FN_i} \quad (2)$$

Average Precision (AP) is a common evaluation method for object detection. AP captures the performance of localization and classification simultaneously. The AP metric score can be computed using equation (3), where $p(i)$ denotes the precision of the predictions [28]. Let $b_i \in \mathbb{R}$ be the box coordinates of the i-th prediction and $s_i \in \mathbb{R}$ is its corresponding classification score. Suppose, it is given an image, an object detector outputs N detected bounding boxes for each category as $\mathcal{B} = \{(b_i, s_i)\}_{i=1}^{N}$. AP metric computes the score of matching those predictions and a set of ground-truth bounding boxes $\mathcal{G}$. The true prediction will be assigned one point the false prediction will be given zero. The positive set $\mathcal{P}$ is the predictions assigned to ground-truth bounding boxes and the other predictions form the negative set $\mathcal{N}$.

$$AP = \frac{1}{|\mathcal{P}|} \sum_{i \in \mathcal{P}} p(i) \quad (3)$$

Intersection over Union (IoU) is used to measure the performance of methods for object category segmentation. IoU measures the similarity between the predicted region and the ground-truth region for an object present in the set of images. IoU can be defined using equation (4) [29], where TP, FP, and FN refer to the number of the true positive, false positive, and false negative, respectively. IoU score lies between 0 and 1, where 0 shows no overlap and 1 is the perfect overlap of prediction and ground truth. Given a threshold α, APα explains that AP is evaluated at IoUα. For instance, AP50 is computed at $IoU = 0.5$. The mean Average Precision (mAP) is defined by equation (5), where n is the number of classes.

$$IoU = \frac{TP}{FP + TP + TN} \quad (4)$$

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i \quad (5)$$

Average Recall (AR) sums up the distribution of recall across a range of overlap thresholds. Average recall between IoU 0.5 and 1 can be computed using equation (6), where gt refers to ground truth [30]. Algorithms with lower AR also have lower mAP and algorithms with high AR also have high mAP.

$$AR = \frac{2}{n} \sum_{i=1}^{n} \max(IoU(gt_i) - 0.5, 0) \quad (6)$$

C. Research workflow

The research workflow is illustrated in Figure 3. In this paper, the pre-processing step is creating bounding boxes to identify the target in the images. It uses LabelImg as a tool for creating the bounding boxes [31]. This paper uses the EfficientDet method from TensorFlow Lite Model Maker. EfficientDet-Lite series are object detection models for mobile and IoT derived from the EfficientDet architecture [32]. The models are trained using a garbage dataset. The trained models are evaluated using a testing dataset and their performance is measured using AP and AR. A trained object detection model is exported to the TensorFlow Lite format *.tflite*. The model is then used as a model for object detection for a mobile application. The mobile application is developed using Android Studio supported by Kotlin. User testing is conducted to test the mobile application.

III. RESULT AND ANALYSIS

A. Data preparation

The original dataset consists of several classes, i.e., green-glass, brown-glass, white-glass, biological, battery, cardboard, metal, paper, plastic, clothes, shoes, and trash is obtained from a public dataset [33]. Green glass, brown glass, and white glass are simplified as one class called glass. The

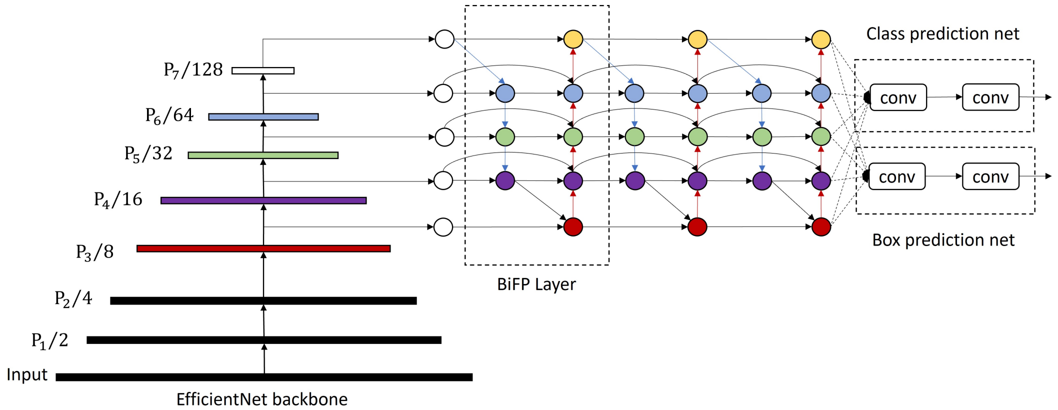

Fig. 2. EfficientDet architecture.

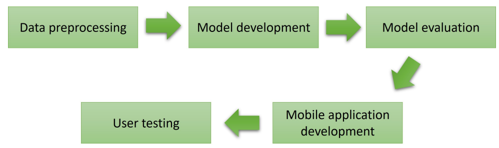

Fig. 3. Research workflow.

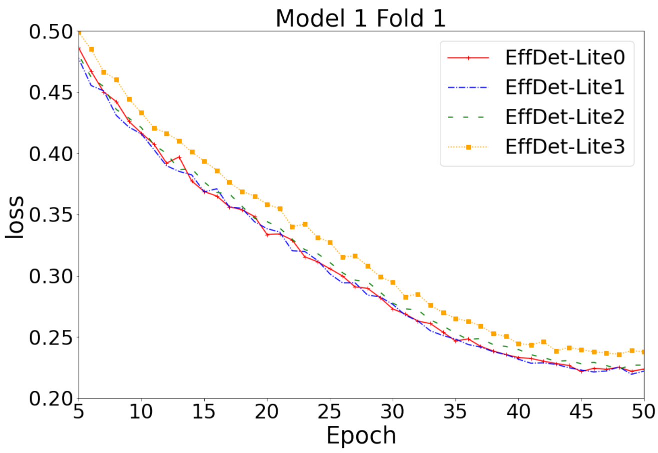

Fig. 4. Train loss of Model 1 Fold 1.

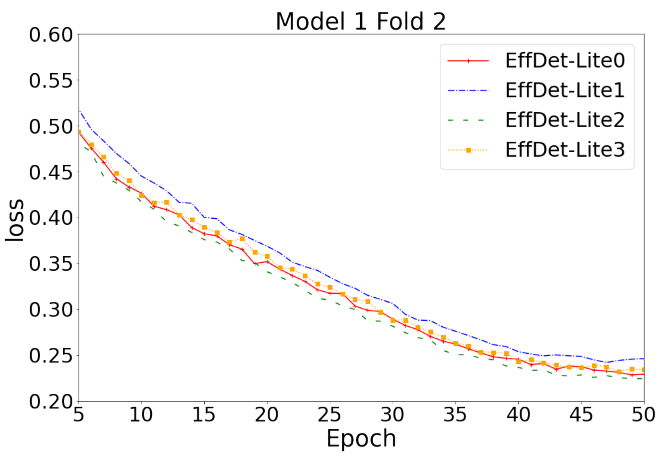

Fig. 5. Train loss of Model 1 Fold 2.

authors add electronic garbage from their own collection so the experiments use 11 classes. The solid wastes are then grouped and labeled as recyclable and non-recyclable,

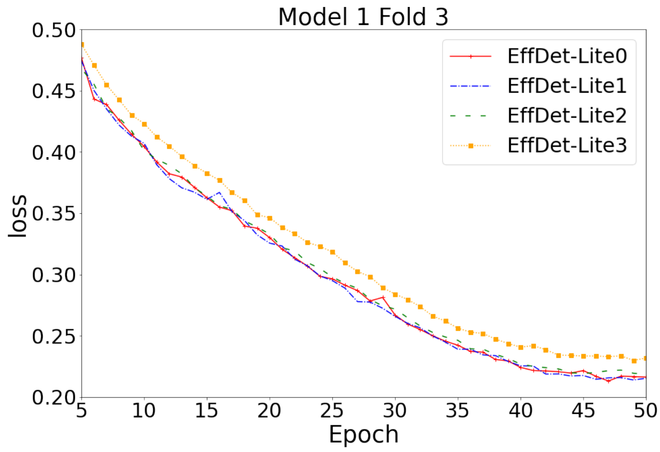

Fig. 6. Train loss of Model 1 Fold 3.

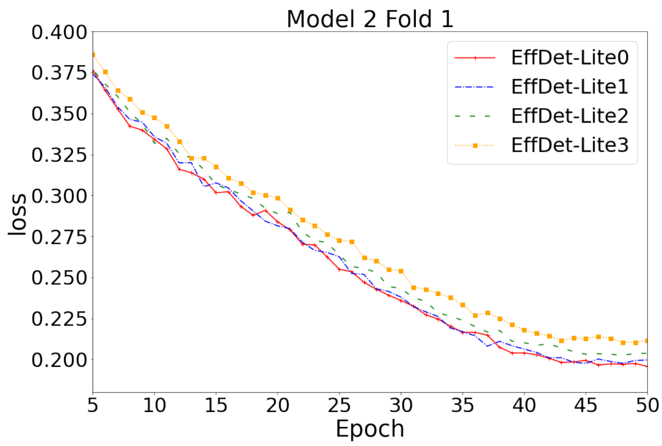

Fig. 7. Train loss of Model 2 Fold 1.

manually.

The dataset is organized using 3-fold cross-validation: Fold 1, Fold 2, and Fold 3. Each fold is divided into $\frac{2}{3}$ training data, $\frac{1}{6}$ validation data, and $\frac{1}{6}$ testing data. The experiments are run in 3 scenarios using datasets of Fold 1, Fold 2, and Fold 3 respectively. Each scenario uses 50 epochs and runs EfficientDet-Lite[0-3] models. The reason to use the EfficientDet-Lite series of 0-3 is to obtain the best model for garbage detection.

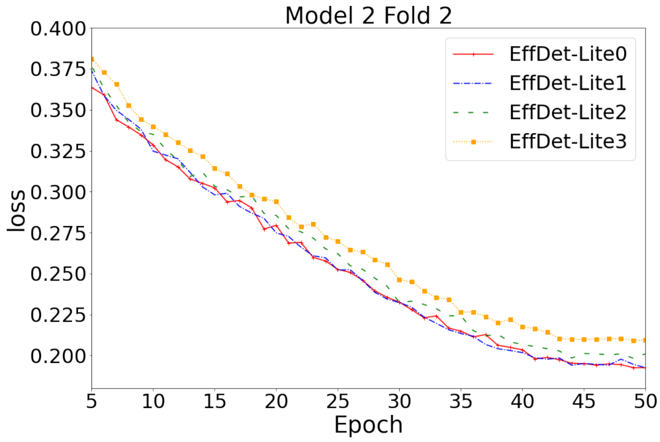

Fig. 8. Train loss of Model 2 Fold 2.

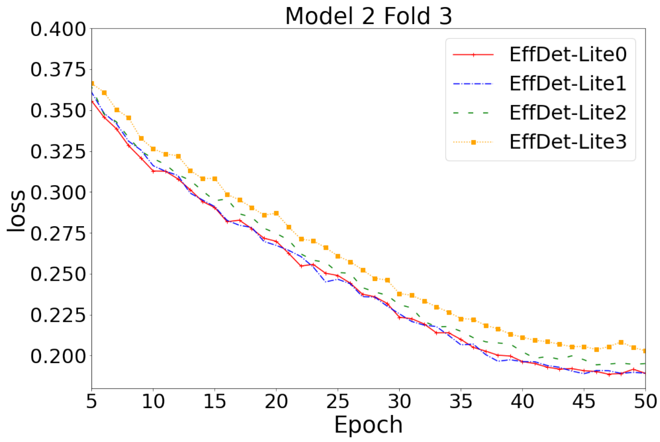

Fig. 9. Train loss of Model 2 Fold 3.

B. Models training and evaluation

The training and evaluation run in Google Colab Python Pro. Figures 4, 5, and 6 show the train loss for Model 1. Training loss results of Model 1 shows that EfficientNet-Lite[0-3] has no significant difference in producing loss values. Figures 7, 8, and 9 show the training loss of Model 2. Generally, the loss values decrease along with increasing the number of epochs. The average time for training is displayed in Figure 10. EfficientDet-Lite0 is less time-consuming and EfficientDet-Lite3 is the most time-consuming.

The models are evaluated using testing data and their performance is measured using AP and AR. The detection precision or recall for large, medium, and small objects is described by letters l, m, and s following the AP and AR, i.e., APl for AP of large objects and so on. Small, medium, and large objects are defined as $area < 32^2$, $32^2 < area < 96^2$, and $area > 96^2$ respectively. ARmax = x refers to AR given x detection per image, where $x = \{1, 10, 100\}$. Table I and Table II show the AP scores from the evaluation step of Model 1 and Model 2. The AR scores of Model 1 and Model 2 are displayed in Table III and Table IV.

The evaluation results of Model 1 can be explained as follows. EfficientDet-Lite3 produces the same average AP50 score as EfficientDet-Lite2 by 0.87. EfficientDet-Lite3 has an average mAP, AP75, and APl of 0.76, 0.83, and 0.79, respectively. Those scores are 0.1 higher than EfficientDet-Lite2 scores. EfficientDet-Lite3 has an average score of APs of 0.47 which is higher than EfficientDet-Lite2 at 0.36. The highest average ARl, ARm, and ARs of Model 1 are achieved by EfficientDet-Lite2 by 0.87, 0.72, and 0.67, respectively. EfficientDet-Lite3 produces lower ARm and ARs by 0.69 and 0.64, respectively. EfficientDet-Lite2 has the highest average of ARm at 0.72 and a higher APm of 0.46 than other models.

The evaluation performance of Model 2 can be elucidated as follows. The average APs scores for EfficientDet-Lite2 and EfficientDet-Lite3 are 0.35 and 0.43, respectively. EfficentDet-Lite3 produces average mAP, AP50, AP75, APl, and APm as 0.79, 0.91, 0.86, 0.82, and 0.55, respectively. Those scores are 0.1 higher than the scores produced by EfficientDet-Lite2. The average score of ARs for EfficentDet-Lite3 is 0.72. Meanwhile, EfficientDet-Lite2 only produces an average ARs score of 0.66. EfficientDet-Lite3 has average scores of ARl and ARm at 0.89 and 0.76 which are 0.1 higher than the scores of EfficientDet-Lite2.

Generally, EfficientDet-Lite2 and EfficientDet-Lite3 show better performance than EfficientDet-Lite0 and EfficientDet-Lite1. EfficientDet-Lite[0-3] produces AP and AR scores with no significant difference, except when detecting the small object. EfficientDet-Lite3 produces slightly higher average AP and AR scores and outperforms other versions to detect small objects. All methods for Model 1 and Model 2 are less accurate to detect small objects due to their average APs scores of 0.35 - 0.47 and average ARs scores of 0.6 - 0.72. The ARmax scores indicate that the more objects in an image, the models perform more accurately. Model 1 and Model 2 have average scores of more than 0.82 for ARmax = 10 and ARmax = 100. Based on the experimental results of Model 1 and Model 2, EfficientDet-Lite3 shows a slightly better performance than others in detecting large objects. It is true that EfficientDet-Lite3 a little bit outperforms but the training process is much slower than EfficientDet-Lite2. Considering the performance and running time, EfficientDet-Lite2 is the recommended model when working with limited resources.

C. User testing

User testing involves 20 participants at the age of over 17 years old. The participants are conducted to capture any kinds of waste using the application Leboh 2 that has been installed on their mobile phones. Figure 11 shows examples of user testing for Model 1. The application of Model 1 classifies the waste into 11 classes: biological (*organik*), battery (*baterai*), cardboard (*kardus*), metal (*logam*), paper (*kertas*), plastic (*plastik*), textile (*tekstil*), shoes (*sepatu*), glass (*beling*), electronic (*electronic*), and trash (*rongsok*). Note that the terms inside the bracket in italics are Indonesian translations for garbage classes. The participants collected 200 screenshots for Model 1 and 200 screenshots for Model 2. The screenshots are evaluated manually. User testing for Model 1 obtains an accuracy, precision, recall, and F1-score of 78%, 82%, 78%, and 79%, respectively. User testing for Model 2 produces an accuracy, precision, recall, and F1-score of 82%, 82%, 82%, and 82%, respectively. A confusion matrix of user testing for Model 1 is displayed in Figure 13 (A). The wrong classifications happened between glass and plastic as 5 items. It is caused by the texture of white glass and plastic that look similar.

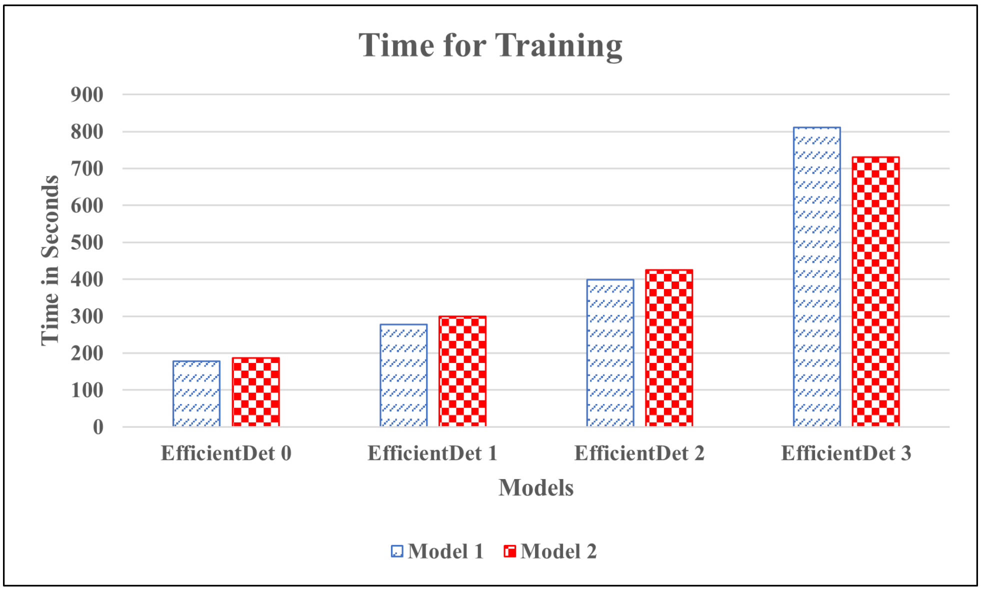

Fig. 10. Time for training

TABLE I
AVERAGE PRECISION (AP) SCORE OF MODEL 1

Dataset	Models	mAP	AP50	AP75	APl	APm	APs
Fold 1	EfficientDet-Lite0	0.75	0.87	0.82	0.78	0.38	0.26
	EfficientDet-Lite1	0.76	0.87	0.82	0.78	0.43	0.33
	EfficientDet-Lite2	0.76	0.88	0.83	0.79	0.41	0.21
	EfficientDet-Lite3	0.77	0.88	0.84	0.80	0.41	0.44
Fold 2	EfficientDet-Lite0	0.74	0.87	0.82	0.77	0.42	0.52
	EfficientDet-Lite1	0.74	0.87	0.81	0.77	0.45	0.67
	EfficientDet-Lite2	0.76	0.88	0.84	0.79	0.46	0.63
	EfficientDet-Lite3	0.77	0.88	0.84	0.80	0.45	0.66
Fold 3	EfficientDet-Lite0	0.72	0.84	0.79	0.75	0.41	0.26
	EfficientDet-Lite1	0.73	0.85	0.80	0.76	0.51	0.25
	EfficientDet-Lite2	0.75	0.86	0.81	0.78	0.48	0.31
	EfficientDet-Lite3	0.75	0.85	0.81	0.78	0.49	0.30

TABLE II
AVERAGE PRECISION (AP) SCORE OF MODEL 2.

Dataset	Models	mAP	AP50	AP75	APl	APm	APs
Fold 1	EfficientDet-Lite0	0.78	0.91	0.86	0.80	0.47	0.42
	EfficientDet-Lite1	0.77	0.91	0.85	0.80	0.48	0.28
	EfficientDet-Lite2	0.79	0.91	0.86	0.81	0.51	0.19
	EfficientDet-Lite3	0.80	0.92	0.87	0.82	0.51	0.46
Fold 2	EfficientDet-Lite0	0.77	0.91	0.86	0.80	0.56	0.63
	EfficientDet-Lite1	0.78	0.91	0.86	0.81	0.55	0.70
	EfficientDet-Lite2	0.78	0.91	0.86	0.81	0.55	0.64
	EfficientDet-Lite3	0.80	0.92	0.87	0.82	0.58	0.66
Fold 3	EfficientDet-Lite0	0.74	0.87	0.82	0.78	0.49	0.12
	EfficientDet-Lite1	0.76	0.88	0.83	0.80	0.51	0.11
	EfficientDet-Lite2	0.76	0.88	0.84	0.80	0.55	0.15
	EfficientDet-Lite3	0.77	0.89	0.85	0.81	0.57	0.16

The example of user testing for Model 2 is displayed in Figure 12. This model is used to detect whether an item is recyclable (*daur ulang*) or not recyclable (*tidak daur ulang*). The top part in Figure 12 from left to right is a glass bottle (recyclable/*daur ulang*), cardboard (recyclable/*daur ulang*), a plastic bag (not recyclable/*tidak daur ulang*), a plastic bottle (recyclable/*daur ulang*), and a metal lid (recyclable/*daur ulang*). Meanwhile, the bottom part shows toothbrushes (not recyclable/*tidak daur ulang*), plastic wrappers (not recyclable/*tidak daur ulang*), styrofoam (not recyclable/*tidak daur ulang*), and watermelon peel (not recyclable/*tidak daur ulang*). Figure 13 (B) shows the confusion matrix of user

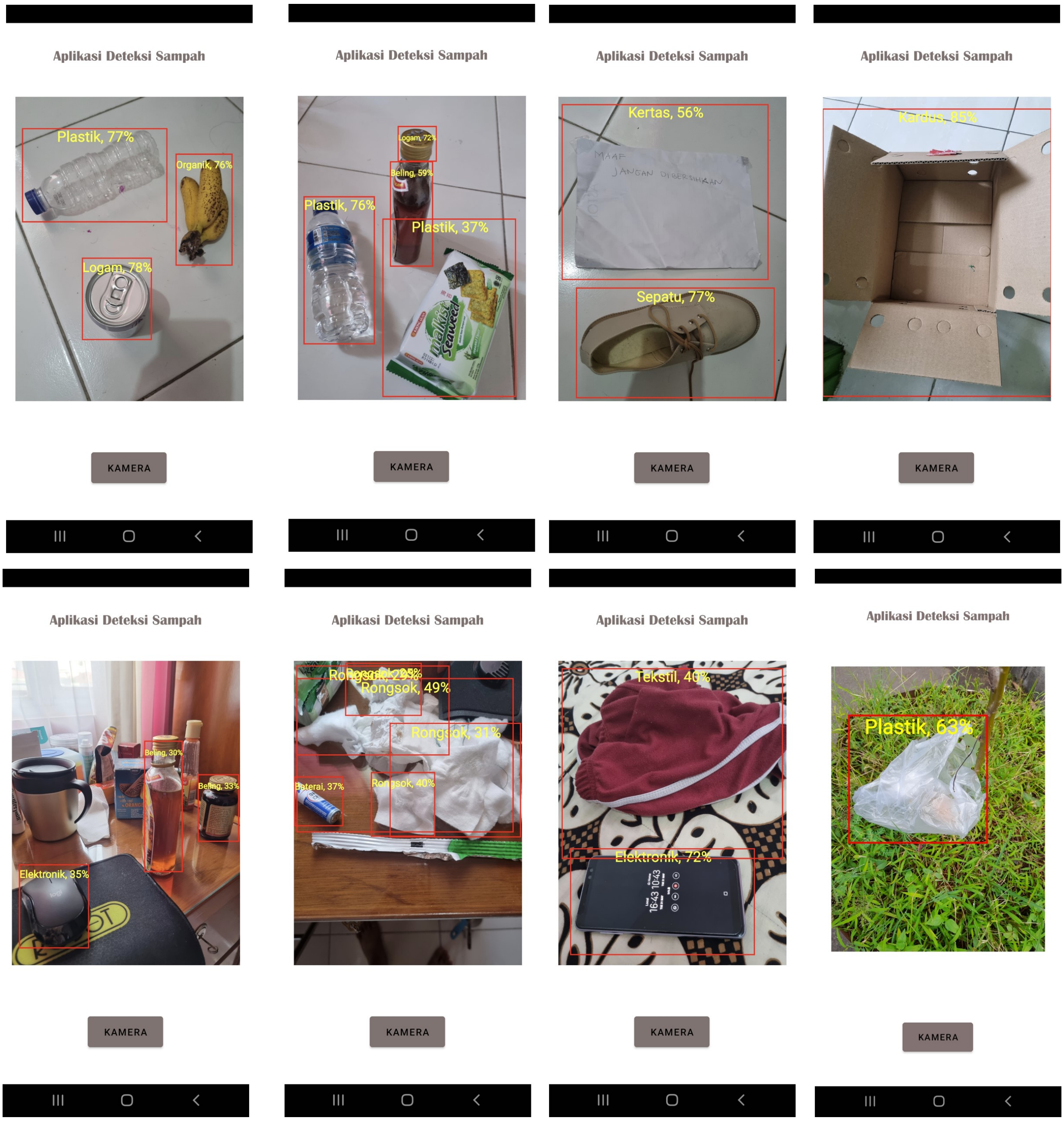

Fig. 11. The example of user testing for Model 1.

TABLE III
AVERAGE RECALL (AR) SCORE OF MODEL 1

Dataset	Models	ARl	ARm	ARs	ARmax = 1	ARmax = 10	ARmax = 100
Fold 1	EfficientDet-Lite0	0.86	0.61	0.53	0.76	0.84	0.85
	EfficientDet-Lite1	0.87	0.65	0.73	0.76	0.85	0.85
	EfficientDet-Lite2	0.87	0.70	0.75	0.76	0.85	0.86
	EfficientDet-Lite3	0.88	0.67	0.76	0.77	0.86	0.87
Fold 2	EfficientDet-Lite0	0.86	0.67	0.75	0.72	0.83	0.84
	EfficientDet-Lite1	0.86	0.67	0.80	0.72	0.83	0.84
	EfficientDet-Lite2	0.87	0.70	0.79	0.73	0.84	0.86
	EfficientDet-Lite3	0.87	0.72	0.80	0.74	0.85	0.86
Fold 3	EfficientDet-Lite0	0.85	0.64	0.47	0.68	0.81	0.83
	EfficientDet-Lite1	0.85	0.72	0.40	0.70	0.83	0.84
	EfficientDet-Lite2	0.86	0.75	0.48	0.70	0.83	0.85
	EfficientDet-Lite3	0.86	0.71	0.36	0.70	0.84	0.85

Fig. 12. The example of user testing for Model 2.

TABLE IV
AVERAGE RECALL (AR) SCORE OF MODEL 2

Dataset	Models	ARl	ARm	ARs	ARmax = 1	ARmax = 10	ARmax = 100
Fold 1	EfficientDet-Lite0	0.88	0.72	0.78	0.74	0.85	0.86
	EfficientDet-Lite1	0.88	0.72	0.74	0.75	0.85	0.86
	EfficientDet-Lite2	0.88	0.74	0.72	0.75	0.86	0.87
	EfficientDet-Lite3	0.89	0.75	0.84	0.76	0.87	0.88
Fold 2	EfficientDet-Lite0	0.88	0.74	0.76	0.70	0.84	0.86
	EfficientDet-Lite1	0.88	0.77	0.80	0.70	0.85	0.86
	EfficientDet-Lite2	0.88	0.76	0.76	0.70	0.85	0.87
	EfficientDet-Lite3	0.89	0.76	0.83	0.71	0.86	0.87
Fold 3	EfficientDet-Lite0	0.86	0.71	0.27	0.66	0.82	0.83
	EfficientDet-Lite1	0.87	0.75	0.42	0.67	0.83	0.84
	EfficientDet-Lite2	0.87	0.76	0.50	0.68	0.83	0.85
	EfficientDet-Lite3	0.88	0.77	0.50	0.68	0.84	0.86

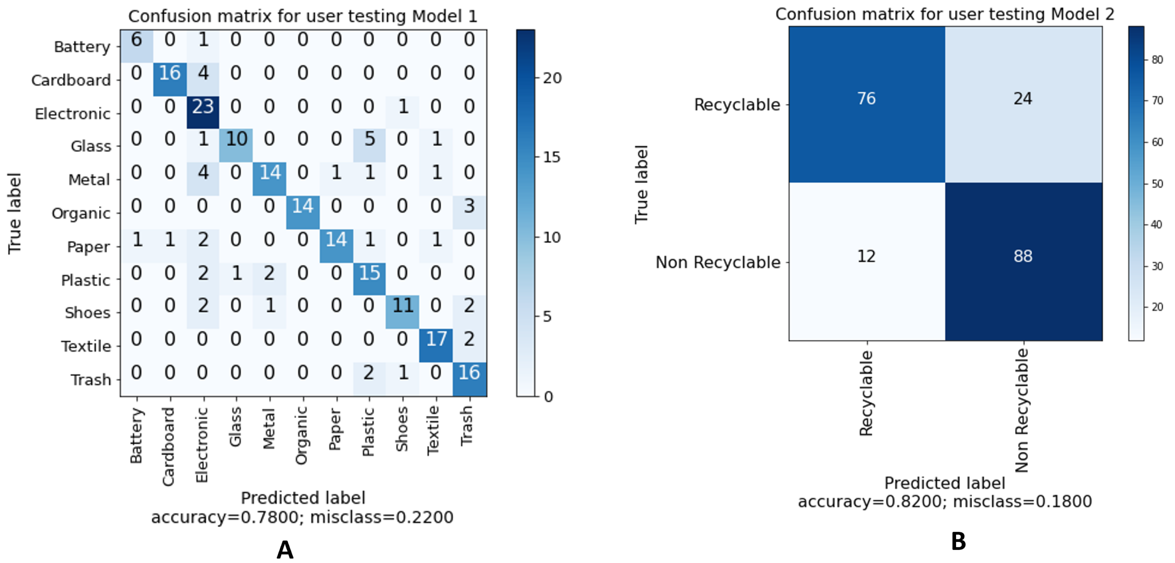

Fig. 13. Confusion Matrices for user testing Model 1 and Model 2.

testing for Model 2. The average misclassification is 18%. The percentage of recyclable that is misclassified is 24% and the percentage of not-recyclable being misclassified is 12%.

The difference between the mAP scores (in-lab evaluation) and precision scores of user testing for all models is no more than 15%. In-lab refers to experiments using Google Colab Python. In-lab experiments and user testing produce average precision scores of over 70% for medium and large objects. It implies that the gap between in-lab model testing and user testing is still acceptable. In-lab model testing produces more detailed evaluation scores for large, medium, and small objects. The testing by users does not distinguish the size of objects. Leboh 2 has been successfully developed to recognize several types of garbage in an image and it fixed the lack of Leboh.

A simple survey was conducted on 172 respondents to obtain information about their opinion on sorting waste. The respondents are Indonesian and living in this country. The respondents receive questions in Google form via social media. They are in the age of over 17 years old to make sure that they have enough experience to explore their living environment and public facilities. The distribution of respondents' age is 72.1% less than equal to 30 years old. The survey results are explained in Table V. The information about respondents' habits in sorting waste supports the idea of developing a mobile application for waste detection as a tool that helps users sort their garbage. The amount of 94.7% of respondents agreed that sorting waste is important to do. The bins for general waste are available in almost all public facilities. However, the number of bins for categorical waste, e.g. recyclable and non-recyclable bins, landfill bins, and cigarette bins, is limited or not available in their living environment.

TABLE V
THE SURVEY RESULTS

Topic	Yes	No
The availability of public bins in the neighborhood.	62.8%	37.2%
The availability of bins in the public facilities, i.e. markets, bus and train stations, and public heath centers.	81.4%	18.6%
The availability of specific bins, e.g, recycled and not recycled bins, batteries bins, electronic bins, dog waste bins, and cigarette bins.	61.0%	39.0%
The respondents have been learning how to sort the waste.	77.3%	22.7%
The respondents are willing to sort the waste before putting it into the proper bins.	96.5%	3.5%
The respondents used to sort the waste according to the materials, e.g, plastics, metal, organic, paper, cardboard, textile, glass, electronics, batteries, and trash	47.7%	52.3%
The respondents used to sort the waste according to the types, i.e. recycled waste and not recycled waste.	37.2%	62.8%

IV. Conclusion

In conclusion, EfficientDet-Lite works well as a model for garbage detection in Android applications. The experimental results of EfficientDet-Lite3 in Google Colab Python obtain mAP are around 75% - 77%. User testing for the first model that is trained to recognize 11 types of garbage produces an accuracy of 78%. The second model is used to detect whether garbage belongs recyclable or not recyclable produces mAP for EfficientDet-Lite3 is around 87% - 92% and the user testing has an accuracy of 82%. This paper also reports a quick survey from 172 respondents that reveals some important information about the difficulties of finding specific bins in public areas and the habits of sorting waste. This application is expected as a useful tool for garbage sorting. For future research, the models will be used to develop an automatic machine for garbage selection.

ACKNOWLEDGMENT

The authors would like to thank all users who participate in testing the application and respondents of the survey

REFERENCES

[1] M. Yang and G. Thung, "Classification of Trash for Recyclability Status," http://cs229.stanford.edu/proj2016/report/ThungYang-ClassificationOfTrashForRecyclabilityStatus-report.pdf, 2016.

[2] G. Mittal, K. B. Yagnik, M. Garg, and N. C. Krishnan, "SpotGarbage: Smartphone App to Detect Garbage using Deep Learning," ser. ACM International Joint Conference on Pervasive and Ubiquitous Computing. Heidelberg: ACM, 2016, pp. 940–945.

[3] M. Satvilkar, "Image Based Trash Classification using Machine Learning Algorithms for Recyclablity Status," Master's thesis, School of Computing National College of Ireland, Dublin, 2018.

[4] A. N. Kokoulin, A. I. Tur, and A. A. Yuzhakov, "Convolutional Neural Networks Application in Plastic Waste Recognition and Sorting," ser. Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus). Moscow and St. Petersburg: IEEE, 2018, pp. 1094–1098.

[5] J. Bobulski and M. Kubanek, "Waste Classification System using Image Processing and Convolutional Neural Networks," in *Advanced in Computation Intelligence*, ser. International Work-Conference on Artificial Neural Networks. Springer, 2019, pp. 350–361.

[6] S. Frost, B. Tor, R. Agrawal, and A. G. Forbes, "Compostnet: An image classifier for meal waste," in *Proceeding of IEEE Global Humanitarian Technology Conference (GHTC)*. Seattle, WA: IEEE, 2019.

[7] A. H. Vo, L. H. Son, M. T. Vo, and T. Le, "A novel framework for trash classification using deep transfer learning," *IEEE Access*, vol. 7, pp. 178 631–178 639, 2019.

[8] P. Nowakowski and T. Pamuła, "Application of Deep Learning Object Classifier to Improve e-Waste Collection Planning," *Waste Management*, vol. 109, pp. 1–9, 2020.

[9] C. Shi, C. Tan, T. Wang, and L. Wang, "A Waste Classification Method Based on a Multilayer Hybrid Convolution Neural Network," *Applies Sciences*, vol. 11, no. 8572, pp. 1–19, 2021.

[10] Y. Wu, X. Shen, Q. Liu, F. Xiao, and C. Li, "A garbage detection and classification method based on visual scene understanding in the home environment," *Complexity*, vol. 2021, pp. 1–14, 2021. [Online]. Available: https://doi.org/10.1155/2021/1055604

[11] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, 2015.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 346–361.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf

[15] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, p. 379–387.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37.

[17] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[18] Q. Chen and Q. Xiong, "Garbage classification detection based on improved YOLOV4," *Journal of Computer and Communication*, vol. 8, no. 12, pp. 285–294, 2020.

[19] A. I. Middya, D. Chattopadhyay, and S. Roy, "Garbage detection and classification using Faster-RCNN with Inception-V2," in *IEEE 18th India Council International Conference (INDICON)*, 1-6, Ed., 2021, pp. 16–27.

[20] D. Patel, F. Patel, S. Patel, N. Patel, D. Shah, and V. Patel, "Garbage detection using advanced object detection techniques," in *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, 2021, pp. 526–531.

[21] S. Majchrowska, A. Mikołajczyk, M. Ferlin, Z. Klawikowska, M. A.Plantykow, A. Kwasigroch, and K. Majek, "Deep learning-based waste detection in natural and urban environments," *Waste Management*, vol. 138, pp. 274–284, 2022.

[22] M. Córdova, A. Pinto, C. C. Hellevik, S. A. Alaliyat, I. A. Hameed, H. Pedrini, and R. S. Torres, "Litter detection with deep learning: A comparative study," *Sensors*, vol. 548, no. 2, pp. 1–19, 2022.

[23] D. Ziouzios, N. Baras, V. Balafas, M. Dasygenis, and A. Stimoniaris, "Intelligent and real-time detection and classification algorithm for recycled materials using convolutional neural networks," *Recycling*, vol. 7, no. 1, pp. 1–14, 2022.

[24] S. Geetha, J. S. I. Dasgupta, R. Bera, I. A. Lawal, and S. Kadry, "Design of waste management system using ensemble neural networks," *Designs*, vol. 6, no. 2, pp. 1–19, 2022.

[25] T. Handhayani and J. Hendryli, "Leboh: An android mobile application for waste classification using tensorflow lite," in *Intelligent Systems and Applications*. Cham: Springer International Publishing, 2023.

[26] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 778–10 787.

[27] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[28] T. Chenxin, Z. Li, X. Zhu, G. Huang, Y. Liu, and j. dai, "Searching parameterized ap loss for object detection," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 22 021–22 033. [Online]. Available: https://proceedings.neurips.cc/paper/2021/file/b9009beb804fa097c04d226a8ba5102e-Paper.pdf

[29] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *Advances in Visual Computing*, G. Bebis, R. Boyle, B. Parvin, D. Koracin, F. Porikli, S. Skaff, A. Entezari, J. Min, D. Iwai, A. Sadagic, C. Scheidegger, and T. Isenberg, Eds. Cham: Springer International Publishing, 2016, pp. 234–244.

[30] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 814–830, 2016.

[31] Tzutalin, "Labelimg." [Online]. Available: https://github.com/tzutalin/labelImg

[32] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[33] Mohamed, Mostafa, "Garbage classification (12 classes)," 2020. [Online]. Available: https://www.kaggle.com/datasets/mostafaabla/garbage-classification