YOLO-Green: A Real-Time Classification and Object Detection Model Optimized for Waste Management

Wesley Lin
Department of Science
Morrison Academy Taichung
Taichung, Taiwan
linw@mca.org.tw

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671821

Abstract—Deep neural networks (DNNs) play an important role in our daily lives, from aiding us in menial tasks to solving world issues such as cancer cell detection. However, few pieces of research have been conducted using DNNs and deep learning models as a medium to help classify and detect trash, in efforts to solve our global waste crisis. This is because current DNNs struggle to be both efficient and accurate while detecting indistinct objects such as waste. To address this issue, this work focuses on YOLO-Green, a novel real-time object detection model designed specifically for trash detection. The model is trained on a dataset gathered from real-world trash divided into seven of the most common types of solid waste. With only 100 epochs of training, YOLO-Green achieves an outstanding mAP of 78.04%, FPS of 2.72, while retaining a model size of only 117 MB. Based on the original object detection of YOLOv4, YOLO-Green exceeds YOLOv4 and other popular deep learning models in both its accuracy and efficiency, while maintaining a relatively small model size. Ultimately, this study sheds a positive light on the potential of using deep learning models as an alternative to manual waste management.

Keywords—**Image classification, object detection, deep learning, convolutional neural networks, waste recycling**

I. INTRODUCTION

In the last few decades, rapid urbanization, population growth, and an increase in consumption have led to the growing global waste crisis. As waste accumulates by the day, the traditional method of manual waste management makes solving this global crisis an impossible task. With no alternative, nearly 80% of all trash is left to be accumulated in landfills or sloughing off in the natural environment as litter [1]. These unrecycled solid wastes then directly pollute our environment, threatening both our health and the longevity of our world.

In face of this crisis, there are several potential approaches in reducing waste pollution, such as banning certain types of trash, using more recyclable materials, or re-using products [2]. However, one key approach that will limit waste accumulation is the development of a more efficient method to detect, classify, and recycle waste in conjunction with a better network and vehicle routing for municipal waste collection [3]. The ability to do so will yield a more immediate effect, as the rate of waste production will be closer to the rate of recycling.

Among the different types of waste, solid waste management is arguably the most important. In contrast to other types of waste, solid trash takes up the majority of waste today and often comes directly from humans [4]. Because of its direct interaction with humans, solid waste is the most tangible approach to limit waste pollution. By managing the solid waste humans produce, we can effectively reduce the waste pollution in this world, while gradually reversing the effects the pollution has caused [5].

With the rapid advancement of computer vision, deep learning models can be a very advantageous and applicable solution. Deep learning, a machine learning form, enables computers to learn from experience [6]. During deep learning, machine learning algorithms use multiple processing layers to understand the representation of data with multiple abstraction layers [7]. Through this generalization, deep learning models learn patterns of the input data, which allows it to detect future unseen data. These approaches have allowed researchers to make cutting-edge advancements in solving real-world problems, such as in object and speech recognition, as well as in cybersecurity, medical informatics, and marketing.

Ideally, if deep neural networks are used to detect and classify trash, the recycling of waste will become exponentially more efficient, as waste management will no longer simply rely on manual labor. A deep learning model can serve as an efficient preliminary waste classification system to sort out a complicated mix of trash. In addition, object detection can be used to identify the remaining misplaced trash during the final stage of the waste management process. However, current popular deep learning models such as YOLO,

978-1-6654-3902-2/21/$31.00 ©2021 IEEE

DenseNet, ResNet, and SSD alone are not good fits for trash classification. The indistinct and often varying features of trash cause the detection accuracy and speed to become relatively low.

In this paper, we address these shortcomings by proposing a newly developed deep learning model YOLO-Green, in hopes of making it a prevalent, affordable, accurate, and efficient replacement for the manual waste management method used. YOLO-Green is vaguely based on the original object detection algorithm of YOLOv4 designed to create a better model regarding the detection and classification of solid waste.

Unlike traditional objects with distinct features, different types of solid waste vary in terms of their different shapes, sizes, and colors. To detect these differences, the developed model uses upsampling and downsampling to ensure all objects regardless of size will be boxed. In addition, to retain the speed, a fire module architecture is introduced to the model to reduce parameters. In the following sections, YOLO-Green will be described in more detail.

To further contextualize the results of this study, the results from YOLO-Green are compared with several popular deep learning models, including YOLO-v3, YOLO-v4, ResNet-50, DenseNet-121, and SSD300 as mentioned previously.

Figure 1. Garbage pollution caused by the mismanagement of solid wastes

II. RELATED WORK

There has been a lot of work in creating more efficient and accurate deep learning models, such as YOLO-LITE: a real-time object detection algorithm for non-GPU computers [8]. Among the many popular deep learning models that are being optimized, the YOLO model has consistently been the most recognized for its high accuracy and speed. Consequently, the YOLO model serves as a good basis and point of reference for this study.

In recent years, there have only been a few studies on waste classification, and even fewer in waste detection. The most similar project to this study is 'TrashNet', a study that compares the accuracy of four known deep learning models for trash classification [9]. The dataset from our study uses a portion of the dataset from Trashnet. However, in contrast, our study seeks to develop a new model designed for the classification and detection of solid waste.

A. CNNs

Convolutional neural networks (CNNs) are a class of artificial neural networks and are the main architecture that is used for computer vision. Instead of having fully connected layers, a CNN has convolutions based on the shared-weight architecture of the convolution kernels or filters, where the filter is convolved with different parts of the input to create the output [10]. As a result, the use of the convolution layer allows for relational patterns to be drawn from an input. In addition, a convolution layer typically has fewer weights that need to be learned than a fully connected layer as the filters do not need an assigned weight from every input to every output.

B. YOLOv4

YOLOv4 uses dimension clusters along with anchor boxes to predict its bounding boxes. K-means is also used to generate the 9 clusters for YOLOv4 that best determine the bounding boxes of the dataset before training. To maximize its accuracy, the model also uses the hybrid network of successive 1*1 and 3*3 convolutional layers. However, unlike previous YOLO models such as YOLOv3, between each convolutional layer, YOLOv4 uses CSP (constraint satisfaction problems) connections above with Darknet-53 below as the backbone [11]. This difference contributes greatly to the improvement of the model's mAP, as can be seen later on in the result. Meanwhile, the YOLOv4 retains high real-time performance. However, due to the additional CSP connection, YOLOv4 requires a relatively long processing time.

Figure 3 shows the network structure of YOLOv4 which can be divided into four parts: backbone, dense connection block, spatial pooling block, upsampling, and detection layer. The backbone is composed of concatenated convolution layers for extracting image features. YOLOv4 also uses many convolutional layers with 512 and 1024 convolution filters, which results in a large number of parameters, leading to large storage usage and slow detection speed [12]. This paper focuses on addressing these issues while increasing the accuracy of the model in detecting solid waste.

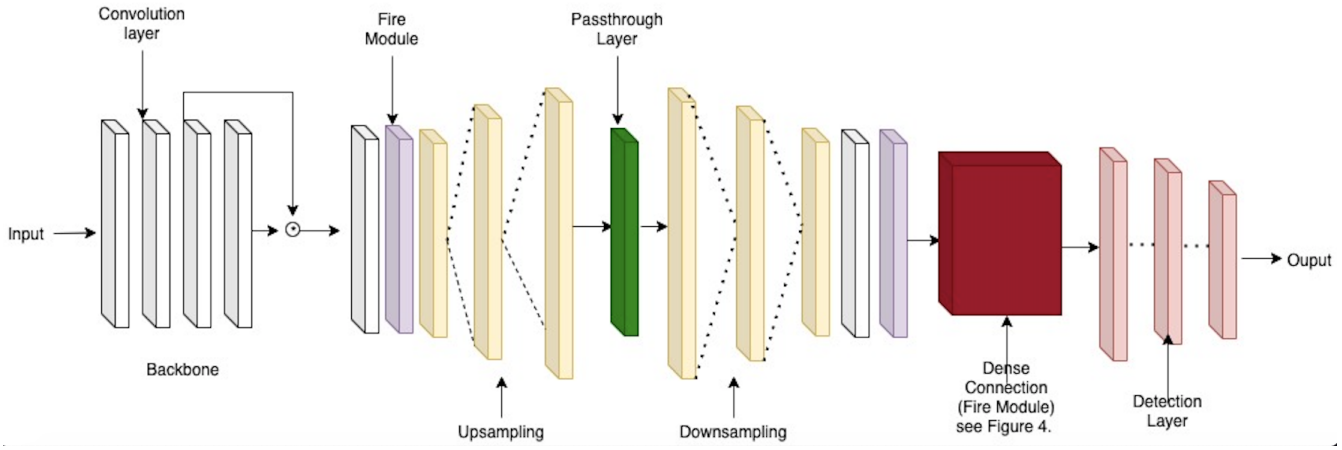


Figure 2. The network structure of YOLO-Green.

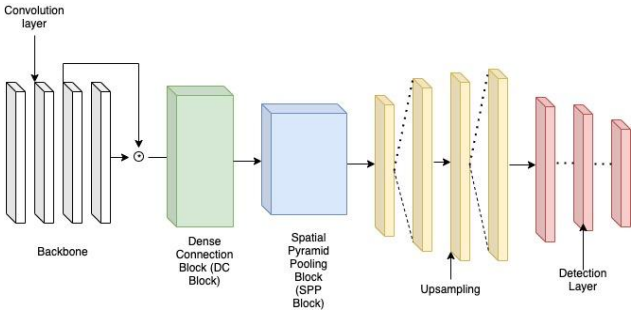


Figure 3. The network structure of YOLOv4

III. YOLO-Green ARCHITECTURE

The overarching architecture of YOLO-Green is inspired by YOLO-v4. Several convolutions were removed within each block from YOLOv4 to reduce the parameters needed to predict bounding boxes. The front part of YOLO-Green retains the front four convolutional layers of YOLOv4. In the middle part, YOLO-Green performs two up samplings, after which YOLO-Green performs two unique downsamplings. A fire module and convolutional layer were added before and after the upsampling and downsampling to reduce the parameters. The model also merges previous feature maps with the passthrough layer before the first detection layer and in between the upsampling and downsampling layer. The last part of the model contains densely connected fire modules and the final detection layers. The overall architecture is illustrated in Figure 2.

A. YOLOv4 in YOLO-Green

The neural network architecture of YOLO has been developed to its fourth version, with improvements that minimize localization errors and increase accuracy. YOLOv4 was selected to be used as the basis for YOLO-Green, notably because of its high accuracy. This is especially important in waste detection, as accuracy is the predominant priority.

Unlike previous models, YOLOv4 uses cross-stage partial connections (CSP) with Darknet-53 as its backbone [13]. In addition, YOLOv4 introduces a new method of data augmentation, called Self-Adversarial Training (SAT). The new data augmentation technique operates in two forward-backward stages. In the first stage, the neural network alters the original image to create the deception that there is no desired object on the image. Afterward, in the second stage, the neural network is trained to detect objects on the modified image in a regular way. YOLOv4 also uses Cross mini-Bath Normalization (CmBN), which allows statistics to only be collected between mini-batches within a single batch [14]. These changes greatly increase the mAP of the model compared to its previous versions.

The YOLO-Green model keeps the first four convolutional layers in YOLOv4 as well as the detection layers at the end of the model. YOLO-Green also employs upsampling to retain fine-grained features. These structural similarities are key components in allowing YOLO-Green to achieve high accuracy.

B. Fire Module in YOLO-Green

In YOLO-Green, the fire module is introduced to reduce the number of parameters, while at the same time increasing the depth and width of the network [15]. The fire module of SqueezeNet uses the bottleneck layer network to compress the model, and the module is widened without decreasing the detection accuracy heavily. In the model, the 1*1 convolutional layer is used to replace the usual 3*3 convolutional layer, which effectively reduces the number of parameters

while achieving similar accuracy. However, we found that if all convolutional layers were replaced by fire modules, the detection accuracy would be decreased significantly. As a result, each fire module is preceded by a darknet 1*1 convolutional layer. We found that the 1*1 convolutional darknet layer serves as a good counterbalance for the fire module, by acting as a sufficient filter without the use for high parameters This balance allows YOLO-Green to attain its high accuracy while reducing its parameter and processing time.

C. Retaining Fine-Grained Features

YOLO-Green employs two structures within its model to retain fine-grained features. After the input image passes through a 1*1 convolutional layer and a fire module, the image is upsampled twice then down-sampled twice. The upsampling ensures that smaller objects could be detected, while the downsampling ensures that larger objects are remembered [16]. The upsampling and downsampling contribute greatly to YOLO-Green's high accuracy as solid waste detection requires focusing on small details, as waste can vary differently and have indistinct features.

In YOLO-Green, the passthrough layer is also utilized to achieve the combination of multi-scale features. By using the passthrough layer, YOLO-Green is able to acquire the upper-level features of 1024*1024 along with the final 512*512 output features. YOLO-Green combines the passthrough layer with the output of the upsampling and downsampling layer, which improves the detection accuracy without bringing in new parameters. The boxes are predicted at two different scales and the pass-through layer is used to take the feature maps from the previous upsample and fire module, similar to YOLOv4. Following this, a 1*1 convolution and fire model layer is added to the process, where eventually, a similar tensor is predicted.

D. Dense Connection between the Fire Modules

A dense connection is employed between the fire modules at the end of the model to improve its accuracy by strengthening feature extraction ability and ensuring maximum information is retained in the network.

In SqueezeNet, fire modules feed forward the output of the nth layers as input to the *n+1th* layer. By employing the DenseNet network, the *nth* fire module of YOLO-Green receives all feature maps from the previous modules in addition to the concatenation of the feature maps [17]. In comparison, this DenseNet network strengthens the feature propagation and improves accuracy.

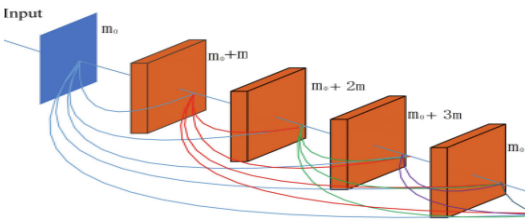


Figure 4. Dense connection between the four fire modules [18]

As shown in Figure 4. The feature maps of the *n-1* fire module are concatenated and used as input for the *nth* fire module. In the figure, m_0 refers to the feature map of the previous convolutional layer, whereas m refers to the current feature map of the fire module. Therefore the nth fire module outputs m_0+m(l-1) feature maps.

If the dense connection is deployed in larger feater or maps, it will result in a large amount of calculation which can affect the real-time processing speed. We tested the real-time performance of different fire modules and determined that four densely connected fire modules achieve the best real-time performance.

IV. EXPERIMENTAL RESULT

A. TrashX & TrashNet Dataset

This work relies on a dataset, referred to as TrashX, as well as images from the TrashNet dataset [19]. The combined dataset is a collection of real-world solid waste images taken on a white background. The various exposure and lighting selected include variations in the dataset. The devices used were Apple iPhone 12 Pro, Apple iPhone 5S, and Apple iPhone SE.

The combined dataset is divided into seven categories of solid waste images retrieved, to account for the seven most common types of solid trash. These categories are batteries, clothes, e-waste, glass, metal, paper, and plastic respectively. Different variations of each category of trash were collected, to ensure a holistic representation of each category. As a whole, the dataset contains 4619 images, with a size of 2.1 GB.

It was determined when comparing trials that reducing the input image size by a factor of one-half can nearly double the speed of the neural network (4.31 FPS vs. 2.72 FPS). Reducing the input size causes less data from each image to pass through the network. Though this allows the network to be relatively leaner, it causes data to be lost. As a result, for our purposes, we determined that it is better to take accuracy (mAP) over speed (FPS)

TABLE 1. Comparison between YOLO-Green and other popular deep learning models

Model	*mAP*	*FPS (Frames per Sec.)*	*Model Size*	*Time*	*Epoch*
YOLOv3	28.83%	1.02	247 MB	16 hrs	100
YOLOv4	45.30%	1.33	257 MB	40 hrs	100
YOLO-Green	**78.04%**	**2.72**	117 MB	12 hrs	100
ResNet-50	16.33%	2.54	146.7 MB	**3 hrs**	100
DenseNet-121	18.25%	1.55	**70.7 MB**	4 hrs	100
SSD300	21.59%	2.01	411.9 MB	7.5 hrs	100

In the experimental portion of the study, 80% of the images were randomly selected for training, while 10% were for validation, and 10% were for testing. The content of the combined dataset is as follows:

- Batteries: 513
- Clothes: 648
- E-waste: 68
- Glass: 692
- Metal: 700
- Paper: 729
- Plastic: 652

Figure 5. Sample images from the dataset

B. Results

All of the experiments in this study were performed using the Keras library version 2.2.4 with Tensorflow version 1.14.0. The deep learning models were trained on a MacBook Pro. Due to the MacBook Pro's weaker GPU in this aspect, the FPS measured can be relatively lower and training time relatively higher than using other commercially available GPUs, such as the Tesla K80 GPU. All models were trained to 100 epochs on the same training, validation, and testing dataset to ensure the fairest comparison between the results.

Training models for the experiments were implemented from scratch: from data training to weights. The values from each model after 100 epochs of training are shown in Table I. The table compares the mAP, FPS, model size, and time required for training for each model.

1) Performance Comparison on mAP

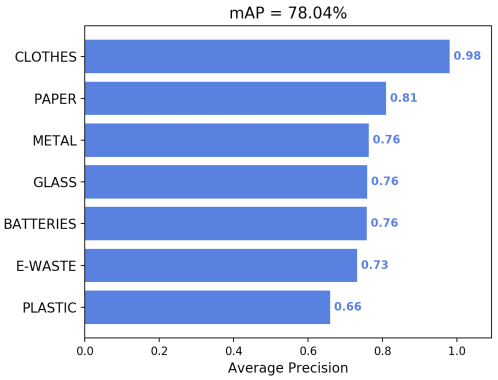
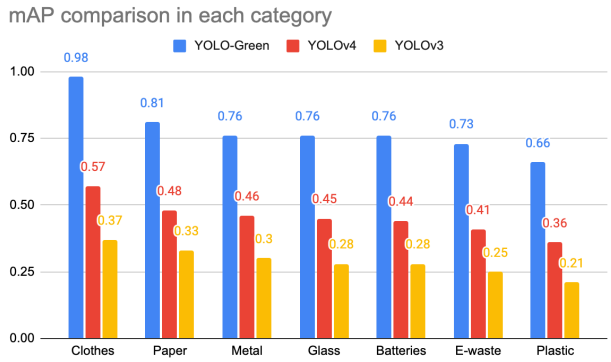


Figure 6. Comparison between mAP in perspective

After training 100 epochs alone, YOLO-Green had an outstanding mAP of 78.04%. This is nearly double the mAP of YOLOv4 and nearly triple the mAP of YOLOv3. The mAP accuracy was also significantly higher than the mAP of the other popular known deep learning models. The densely connected fire module, as well as the upsampling and downsampling, were aspects that greatly contributed to the high accuracy. YOLO-Green's mAP testing result within each category is described below in Figure 6. With the dataset, YOLO-Green is best at detecting clothing waste, and least accurate while detecting plastic. This is understandable because plastic varies greatly in its shape and size, and most disadvantageously is clear in

color. However, an mAP of 66% in detecting plastic is still very accurate.



Model	*Average mAP*
YOLO-Green	**78.04%**
YOLOv4	45.30%
YOLOv3	28.83&

Figure 7. mAP values of three methods on the dataset

Figure 7. provides a detailed comparison between the mAP result for each of the seven categories of waste in YOLO-Green, YOLOv4, and YOLOv4. The values indicate that YOLO-Green is not only more accurate on average, but also more accurate at detecting all of the seven categories of solid waste individually, relative to YOLOv4 and YOLOv3.

2) Model Loss

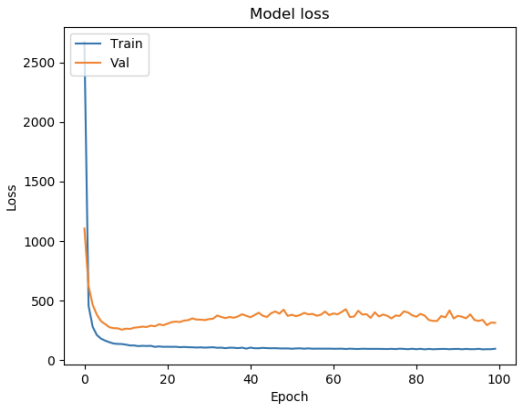


Figure 8. YOLO-Green model loss graph

Figure 8. shows YOLO-Green's model loss curve. In a process called empirical risk minimization, a machine learning algorithm builds a model by examining many examples and attempting to find a model that minimizes loss. The loss refers to the penalty that is imposed by the model for a bad prediction. The degree of the loss indicates how bad the model's prediction was on a single example. Consequently, if the model's prediction is perfect, the loss is zero; otherwise, the loss is greater. By training, the model aims to find a set of weights and biases that have low loss, on average, across all of the images in the dataset. Therefore, the steep decline in both the training and validation loss curve reflects the model's ability to detect accurately despite is minimal training. While the model loss reached a clear overfit around epoch 10-20, indicating that the model has already reached its training peak, the training was continued to 100 epochs in order to provide a fair and consistent point of comparison with other deep learning models.

3) FPS, Model Size, & Training Time

In addition to being significantly more accurate, YOLO-Green has an FPS higher than YOLOv4, YOLOv3, and the other popular deep learning models. Even though the difference is not as drastic as the values in the mAP, YOLO-Green still outperformed the rest of the models with an FPS of 2.72. This is due to the reduced parameters with the removal of convolutional layers in each block and the introduction of the fire module. Similarly, YOLO-Green had one of the smallest model sizes of 117 MB, half in size compared to the YOLOv4 and YOLOv3. This is a result of the reduced layers and parameters as well. YOLO-Green also had a training time of 12 hours, which is less than the training time of both YOLOv3 and YOLOv4. While the model size and training time of YOLO-Green is not the smallest and shortest compared to some of the other popular deep learning models, this is a fair tradeoff as lighter models tend to have significantly lower accuracies. In the four points of comparison, YOLO-Green outperformed YOLOv4 and YOLOv3, in its accuracy, efficiency, and size.

V. APPLICATIONS, IMPLICATIONS, & CONCLUSIONS

In this work, we proposed YOLO-Green which is a real-time classification and object detection system optimized for waste management. YOLO-Green is designed by introducing the fire module from SqueezeNet into YOLOv4 at first to reduce parameters and model size. A densely connected convolutional neural network of fire module, as well as upsampling and downsampling, were implemented to increase accuracy.

The proposed YOLO-Green model achieved an mAP of 78.04%, model size of 117 MB, training time of 12 hours, and 2.72 FPS, surpassing all of its YOLO

counterparts. These results were also significantly better than the results of other popular deep learning models, including ResNet-50, DenseNet-121, and SSD300. In addition, YOLO-Green's accuracy and efficiency in classifying and detecting solid waste can still be increased with a larger dataset and more training.

There are several applications for YOLO-Green, especially if it is integrated into the current manual waste management process. If YOLO-Green is combined with a good computer vision sensor, the model can serve as an efficient preliminary waste classification system or a final detection system for misplaced trash during the waste management process. By integrating this system alongside the current manual waste classification process, the menial manual labor required during this process can be greatly reduced. In sum, the results in this study shed a positive light on the potential of using deep learning models as a cheaper, more accurate, and prevalent alternative to manual trash classification: a closer step toward autonomous waste management.

References

[1] Parker, L. 2018. "A whopping 91% of plastic isn't recycled." National Geographic

[2] Liu, Z; Adams, M; Walker; T, 2018. "Are exports of recyclables from developed to developing countries waste pollution transfer or part of the global circular economy?" In Resources, Conservation and Recycling, 22-23

[3] Beltrami, E. J.; and Bodin, L. D. 1974. Networks and vehicle routing for municipal waste collection. Networks 4(1): 65–94

[4] Hoornweg, Daniel, Bhada-Tata, Perinaz, "What a waste: a global review of solid waste management," World Bank, March 2012.

[5] Adam D. Read, "A weekly doorstep recycling collection, I had no idea we could!": overcoming the local barriers to participation," Resources, Conservation and Recycling, vol. 26, pp. 217-249, June 1999.

[6] O. Delalleau, Y. Bengio. "Shallow vs. deep sum-product networks," Advances in Neural Information Processing Systems, 2011.

[7] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, Edin Muharemagic, "Deep learning applications and challenges in big data analytics," Journal of Big Data, February 2015.

[8] Huang, R.; Pedoeem, J. and Chen, C., 2018. "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers," In IEEE Conference on Big Data

[9, 19] Arda, A.; Seref, A.; Keskin, R.; Kaya, M., and Haciomeroglu, M., 2018. "Classification of TrashNet dataset based on deep learning models," In IEEE International Conference on Big Data

[10] Albawi, S., Mohammed, T. A., & Al-zawi, S. (n.d.). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*. https://doi.org/10.1109/ICEngTechnol.2017.8308186, CNN

[11] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao, "YOLOv4: optimal speed and accuracy of object detection," 2020.

[12] Redmon, J.; Divvala, S; Girshick, R. and Farhadi, A., 2016a, "You only look once: Unified, real-time object detection," In Proceedings of the IEEE conference on computer vision and pattern recognition, 779– 788.

[13] Bochkovskiy, A.; Wang, C., and Liao, H., 2020. "YOLOv4: Optimal Speed and Accuracy of Object Detection."

[14] Zhu, Q.; Zheng, H.; Wang, Y.; Cao, Y., and Guo, S., 2020, "Study on the Evaluation Method of Sound Phase Cloud Maps Based on an Improved YOLOv4 Algorithm," In Sensors

[15, 18] Fang, W.; Wang, L., and Ren, P., 2020. "Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments," In IEEE Access

[16] Yang, W., and Jiachun Z., 2018. "Real-time face detection based on YOLO," In IEEE International Conference on Knowledge Innovation and Invention (ICKII)

[17] Zhu, Y., & Newsam, S. (n.d.). DenseNet for dense flow. *2017 IEEE International Conference on Image Processing (ICIP)*. https://doi.org/10.1109/ICIP.2017.8296389