A YOLO-based Neural Network with VAE for Intelligent Garbage Detection and Classification

Anbang Ye
Zhejiang University-University of Illinois at
Urbana-Champaign Institute
anbang.17@intl.zju.edu.cn

Bo Pang
Zhejiang University-University of Illinois at
Urbana-Champaign Institute
bo.17@intl.zju.edu.cn

Yucheng Jin
Zhejiang University-University of Illinois at
Urbana-Champaign Institute
yucheng.17@intl.zju.edu.cn

Jiahuan Cui
Zhejiang University-University of Illinois at
Urbana-Champaign Institute
jiahuan.cui@intl.zju.edu.cn

ABSTRACT

Garbage recycling is becoming an urgent need for the people as the rapid development of human society is producing colossal amount of waste every year. However, current machine learning models for intelligent garbage detection and classification are highly constrained by their limited processing speeds and large model sizes, which make them difficult to be deployed on portable, real-time, and energy-efficient edge-computing devices. Therefore, in this paper, we introduce a novel YOLO-based neural network model with Variational Autoencoder (VAE) to increase the accuracy of automatic garbage recycling, accelerate the speed of calculation, and reduce the model size to make it feasible in the real-world garbage recycling scenario. The model is consisted of a convolutional feature extractor, a convolutional predictor, and a decoder. After the training process, this model achieves a correct rate of 69.70% with a total number of 32.1 million parameters and a speed of processing 60 Frames Per Second (FPS), surpassing the performance of other existing models such as YOLO v1 and Fast R-CNN.

CCS CONCEPTS

• **Computing methodologies**; • **Machine learning**; • **Applied computing**; • **Computers in other domains**;

KEYWORDS

Intelligent Garbage Detection and Classification, YOLO-based Neural Network, Variational Autoencoder (VAE)

ACM Reference Format:
Anbang Ye, Bo Pang, Yucheng Jin, and Jiahuan Cui. 2021. A YOLO-based Neural Network with VAE for Intelligent Garbage Detection and Classification. In *2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2020), December 24–26, 2020, Sanya, China.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3446132.3446400

1 INTRODUCTION

People are generating more and more waste each year, causing a severe landfill crisis with more than 2.2 billion tons of garbage to be landfilled by 2025 [1]. Deeply troubled by the ever-growing garbage, its detection and classification for recycling purpose has become an urgent need. Therefore, to realize intelligent garbage recycling, we developed a neural network model that is able to accurately distinguish recyclable garbage from non-recyclable waste for potential industrial applications.

There are some enlightening achievements about image detection and classification, one of which is the state-of-the-art You-Only-Look-Once (YOLO) model proposed by J. Redmon et al. [2]. Although this model achieves extraordinary performance on regular image detection and classification tasks, it frequently struggles with locating bounding boxes for oblique objects. Therefore, in this study, we adopted the idea of rotated object detection and constructed a YOLO-based neural network model which is able to obtain the orientation information of the garbage objects with respect to their horizontal axes. Then by applying a Variational Autoencoder (VAE), we improved the accuracy of our model, and made it more robust against over-fitting. Furthermore, to train a reliable model, we produced a special dataset for experiments under the constraint that there is no current available dataset suitable for our scenario. The result of this study is promising, since the model achieves a relatively high correct rate with a faster speed and a smaller model size compared with other existing models such as YOLO v1 and Fast R-CNN. Therefore, we are confident that future researchers could bring our model into the real-world applications such as being deployed on portable, real-time, and energy-efficient edge-computing devices.

The rest of this paper is organized as follows: First, Section II highlights some related work on image detection, classification, and intelligent garbage recycling. Then, Section III describes the methodologies used in this study, including how to prepare the dataset, how to modify the YOLO v1 neural network to make it compatible with our model, and how to implement an effective VAE. Section IV introduces our detailed experimental setup, followed by Section V, the experimental results and analysis with regard to the accuracy, speed, and size of the model, as well as the improvement brought by the VAE and some comparisons between the proposed model and other existing models. Finally, Section VI serves as the conclusion part, where we summarize the results of this study,

ACAI 2020, December 24–26, 2020, Sanya, China

ACM ISBN 978-1-4503-8811-5/20/12…$15.00
https://doi.org/10.1145/3446132.3446400

Table 1: Garbage Categories and Labels

Garbage Categories	Garbage Labels
Interested Category	’cell phone batteries’, ’lithium batteries’, ’nickel cadmium batteries’, ’bottles’, ’glass bottles’, ’pesticide bottles’, ’water bottles’, ’cans’, ’pesticides and cans’, ’ring-pull cans’, ’vacuum cups’, ’plastic cups’, ’circuit boards’, ’tetra packs’
Uninterested Category (Noise)	’sunflower seeds’, ’watermelon seeds’, ’shells’, ’walnut shells’, ’peanut shells’, ’edamame shells’, ’glutinous rice wrapped in bamboo leaves’, ’glass bottles and cans for food and daily necessities’, ’PVC tubes’, ’leaflets’, ’magnets’, ’wet wipes’, ’false eyelashes’, ’corn cobs’, ’band-aids’, ’banana peels’, ’pomelo peels’, ’metal products’, ’lighters’, ’disposable lunch boxes’, ’food wrappers’, ’sockets’, ’medical packages’, ’plastic packages’, ’pharmaceutical packages’

emphasize the strengths and weaknesses of our model, and put forward some future work to be done.

2 RELATED WORK

During recent years, one of the most popular image recognition neural network is the You-Only-Look-Once (YOLO) model. By treating image recognition tasks as regression problems, YOLO utilizes a unified architecture to realize an end-to-end direct optimization on detection performance [2]. As a result, YOLO can perform fast and accurate predictions, and also generalizes well. However, YOLO is not good at highlighting bounding boxes for oblique objects, to solve this problem, an ideal choice is to adopt refinement for detecting rotated objects, as X. Yang et al. [3] proposed in their R^{3}Det model. Inspired by J. Redman et al.’s YOLO and X. Yang et al.’s R^{3}Det, combining their ideas together, it forms the basis of our neural network structure that is detailedly described in Section III.

Before this study, there have been considerable researches on intelligent garbage recycling. G. White et al. [4] designed a CNN-based model, WasteNet, to be deployed on low-power devices at the edge such as garbage bins. Similarly, inspired by G. Thung and M. Yang’s [5] Support Vector Machine (SVM) and CNN-based models, O. Adedeji and Z. Wang [6] formulated another model using ResNet-50 as the extractor and a SVM as the classifier. Apart from these software models, a vivid example of a real-world application is M. U. Sohag and A. K. Podder’s [7] smart gabage management system constructed through the IoT technology. The above researches provide us with an insicive insight into the entire development process of an intelligent garbage recycling system, which lies a solid foundation for our later experiments—throughout this study, we are using these previous works as benchmarks for reference.

3 METHODOLOGY

This section elaborates the methodologies used in our study, including data preparation, the proposed YOLO-based neural network structure, Non-maximum Suppression (NMS), and the implementation of the VAE.

3.1 Data Preparation

Current available datasets pertaining to garbage detection and classification are not designed for recycling purpose [5], which contradicts to our goal of detecting recyclable garbage. In order to trian an effective model for recyclable garbage such as cans, batteries, and plastic bottles, we generated our own dataset for this study based on raw images from the 2020 Haihua AI Challenge (2020 HAC), a garbage sorting competition [8].

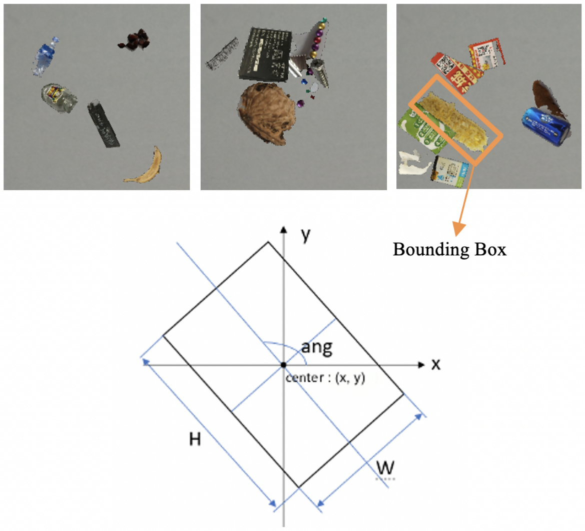

Figure 1: Sample Images and the Layout of the Bounding Box

The whole dataset consists of 124,858 images with resolution of 256 pixels × 256 pixels. These images are divided into three sets: the training set contains 101,138 images, the validation set contains 11,235 images, and the test set contains 12,485 images. To generate images that are representative of the real-world scenario (e.g. scattered garbage on conveyor belts), in our dataset, each image contains 1-3 objects from the interested category (target) and 3-6 objects from the uninterested category (noise). The total number of garbage labels in the interested category is 14, while the total number of garbage labels in the uninterested category is 25. Table 1 gives all the garbage categories and labels.

The positions and orientations of both targets and noises are distributed randomly. For simplicity, all images in our dataset have the same grey background color as shown in Fig. 1. This is because the color of regular conveyor belts is normally uniform and can be eliminated through technological means.

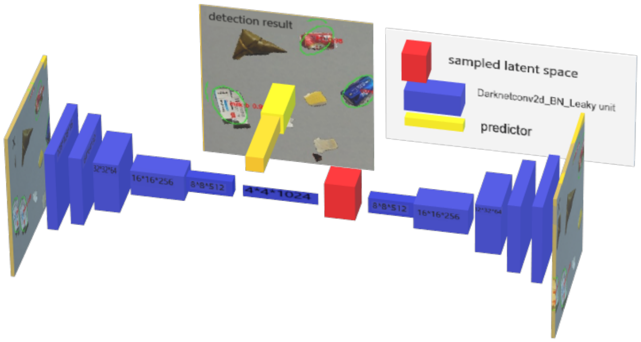

Figure 2: Proposed YOLO-based Neural Network Structure

For better distinction, each image is linked to a label file, which contains basic information about its corresponding image, such as the size of the image and its path. The label file also includes a description of the labels, positions, and orientations of all objects from the interested category in the image. Finally, the bounding box for each target object reflects the x and y coordinates of its center, the angle with respect to the horizontal axis, and the height and width of the bounding box itself.

3.2 YOLO-based Neural Network Structure

The network structure in this study is based on YOLO v1, which is consisted of a convolutional feature extractor, a convolutional predictor, and a decoder. The network divides the input into 4 × 4 grids. The convolutional predictor generates 4 × 4 × 1 sets of outputs for the input image. Each output is an encoding of the confidence, one-hot object label, and bounding box for an object whose center is located in a grid of size 64 × 64 if exists. YOLO v1 originally predicts 7 × 7 × 2 sets of outputs, but we decided to reduce the number of predictors for each image. This is because in the real-world applications, the receptive fields of the network need to be small enough in order to enable robotic arms to move more efficiently with small variation and make inference faster. Another difference from YOLO v1 is that our proposed model predicts only one bounding box for each grid instead of two. This is because during the experiments, we started with a model that predicts two bounding boxes like YOLO v1, but the accuracy turned out to be relatively low. Two reasons may possibly explain this outcome. First, YOLO v1 will highlight two bounding boxes for each grid such that each predictor is specialized in predicting either the larger bounding box or the smaller one. On PASCAL VOC dataset, YOLO v1 can achieve high accuracy by this idea, because objects in PASCAL VOC are either very large or very small. However, in our dataset, objects in the images do not differ much in their sizes, thus making the prediction of two bounding boxes per grid meaningless. Second, the original method doubles the number of negative samples, making the training process more difficult and easier to diverge, forcing YOLO v1 to use some special training techniques, such as to start with small learning rate and slowly increase the learning rate to reach convergence.

Besides, by reducing the number of predictors, we decreased the number of parameters to be trained, thus making the training process faster and easier. Furthermore, because our network increases the proportion of regions that contain objects, the model is able to balance positive and negative samples and facilitates solving the instability problem of the predicted confidence with YOLO [2]. Also, we utilized the Darknetconv2d_BN_Leaky unit from YOLO v3 in the convolutional feature extractor and the predictor, which is consisted of a convolutional layer followed by a batch normalization layer and an activation layer [9]. The basic network structure is a concatenation of five Darknetconv2d-_BN_Leaky units. Without using pooling layers, we greatly improved the inference speed of the network. Finally, similar to YOLO v3, we used leaky ReLU as the activate function for all layers [9].

3.3 Non-maximum Suppression (NMS)

Similar to YOLO v1, Non-maximum Suppression (NMS), a technique that is widely used in selecting one object from several overlapping objects [10], plays an important role in the proposed model. For example, when an object is located in the middle of two grids, the predicted confidences of both grids can be high, such that both grids predict the same label. However, it is also possible that there are two objects with the same label that locate in two separate grids, which may lead to the same result as the previous example. In the first case, a correct prediction should generate only one bounding box, while in the second case, a correct prediction should generate two bounding boxes. Therefore, the model requires a technique to distinguish whether two bounding boxes are attached to the same object or not, and NMS just offers a practical approach to filter out false-positive bounding boxes.

3.4 The Implementation of Variational Auto-encoder (VAE)

One primary novelty of our study is the implementation of a Variational Autoencoder (VAE). VAE is usually considered as an unsupervised generative model, but recently, researchers have shown that using VAE in supervised learning has the potential to improve regression tasks [11], as VAE helps the network to learn interpretable factorized representations of complex read-world data [12]. A vivid example is that Yoo et al. used VAE to improve the regression accuracy of human pose detection of visual data on complex manifold [13].

Assume $\boldsymbol{I} = \{\boldsymbol{i}^{(1)}, \ldots, \boldsymbol{i}^{(\mathrm{n})}\}$ is the training dataset with n images, and each image $\boldsymbol{i}$ is represented by a set $\boldsymbol{O} = \{\boldsymbol{o}^{(1)}, \ldots, \boldsymbol{o}^{(\mathrm{m})}\}$ consisted of m objects, corresponding to their spacial information and categories $\boldsymbol{C} = \{\boldsymbol{c}^{(1)}, \ldots, \boldsymbol{c}^{(\mathrm{m})}\}$. Besides $\boldsymbol{C}$, other trivial information about the objects (e.g. colors) is expressed as $\boldsymbol{R} = \{\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(\mathrm{m})}\}$. For each object $\boldsymbol{o}$ in an image, all information associated with it is $\boldsymbol{g} = \{\boldsymbol{c}, \boldsymbol{r}\}$. Assume $\boldsymbol{g}$ is predictable by the neural network and the input image $\boldsymbol{i}$ can be reconstructed from $\boldsymbol{g}$ through the decoder. Furthermore, another predictor is used to extract $\boldsymbol{c}$ from $\boldsymbol{z}$, where $\boldsymbol{z}$ is the latent representation containing all information about the image and is the output of the encoder. The encoder is represented as $P(\boldsymbol{z}|\boldsymbol{o})$, the decoder is represented as $P(\boldsymbol{o}|\boldsymbol{z})$, and the predictor is represented as $P(\boldsymbol{c}|\boldsymbol{g}) = P(\boldsymbol{c}|\boldsymbol{g}) \bullet P(\boldsymbol{g}|\boldsymbol{z})$, which is used to extract all useful information from the latent space $\boldsymbol{z}$. Assume $\boldsymbol{z}$ obeys the

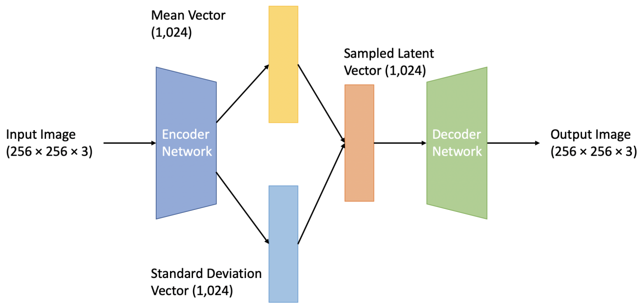

Figure 3: The Variational Autoencoder (VAE) for Each Grid

Gaussian distribution for each object, so we can construct a new sampled latent space $\boldsymbol{z}' = P(\boldsymbol{z}|\boldsymbol{o}) \sim N(\boldsymbol{z}; \mu, \sigma)$, where μ is the mean vector and σ is the standard deviation vector predicted from $\boldsymbol{z}$ using a fully connected layer. Meanwhile, to prevent divergence, further constraints are added to ensure the sampled latent space $\boldsymbol{z}'$ obey the Gaussian distribution $N(0,1)$. Therefore, the loss of our model mainly consists of three parts, the loss of spacial information same as YOLO, the reconstruction loss, and the KL divergence DKL = $(P(\boldsymbol{z}'|\boldsymbol{o}) \,||\, N(0,1))$.

The implementation of the VAE in our proposed model is detailed as follows: a decoder is connected to 16 sampled latent vectors (one latent vector for each grid), which is given by the last layer of our feature extractor using a single Gaussian prior; a fully-connected layer is used to learn the mean vector and the variance vector of a gaussian distribution for each latent vector. The generated mean and variance vectors for each grid is then used to generate a 1,024-dimension sampled latent vector according to Gaussian distribution—the feature extractor, latent vectors, and decoder collectively form a standard VAE network, as shown in Fig.3. Such structure helps us to improve the accuracy of regression and solve the over-fitting problem [11].

4 EXPERIMENTAL SETUP

We implemented the YOLO-based model using TensorFlow backend. During the training process, the number of parameters to be trained was 32.1 million. We trained the model on an Intel Xeon Gold 5120 CPU, and the entire training process lasted for more than 22 hours. Our model was trained for 3 sperate stages: first, we only trained the VAE and left other weights untrained for 40 epochs; second, we only trained the predictor and left the first 5 layers untrained for 50 epochs; third, we trained the global model to make each part reconcile better with each other for 200 epochs. Furthermore, the learning rate l_r was halved every 50 epochs during the training process.

Typically, a VAE contains three parts: an encoder, latent vectors, and a decoder. For encoder and decoder, we applied the structure used in Q. Zhao et al.'s experiments [11], which has been proved to be highly effective. For latent vectors, after several trials, we discovered that a 1,024-diminsion sampled latent vector performs the best with regard to garbage detection and classification, because it is able to contain the garbage information while prevent latent space from getting too sparse.

5 EXPERIMENTAL RESULTS AND ANALYSIS

This section demonstrates the experimental results followed by our analysis of the performance of the proposed YOLO-based model with respect to some existing models for rotated object detection and real-time object detection. Because we wanted to implement the YOLO-based model in real-time detectors that are small enough to be deployed on IoT devices, therefore, we compared our model against other models by the following metrics: accuracy measured by five levels (Correct/Localization/-Similar/Other/Background) defined in J. Redmon and his colleagues' work [2], speed measured by Frames Per Second (FPS), and model size measured by the number of parameters. As a result, we found the proposed model is able to obtain relatively better results than other existing models. Furthermore, we also compared the performance between the model with VAE and the same model without VAE.

5.1 Detection and Classification Results

Fig. 4 shows the results obtained from our proposed model, images in the first row are samples of good results and images in the second row are samples of poor results. Notice that bounding boxes are highlighted in green ellipses and confidences are indicted in red texts.

5.2 Accuracy

Based on the assessment criteria proposed by J. Redmon et al. [2], we can evaluate the performance of a model according to the following three factors: first, whether the model gives a correct classification result on some object; second, the Intersection over Union (IOU) of the predicted bounding box over the ground truth; third, the difference in orientation. From good to poor, we set five levels and our assessment criteria are listed below,

- *i: Correct*: the model classifies correct class, IOU > 0.5, and the difference in orientation < 30{°}
- *ii: Localization*: the model classifies correct class, 0.1 < IOU < 0.5, and the difference in orientation < 30{°}
- *iii: Similar*: the model classifies similar class, IOU > 0.1, or the difference in orientation > 30{°}
- *iv: Other*: the model classifies wrong class, IOU > 0.1, or the difference in orientation > 30{°}
- *v: Background*: IOU < 0.1 for any object

The results of our YOLO-based model and other existing models are given in Fig. 5 and Table 2

Compared with the performance of YOLO v1 and Fast R-CNN, the YOLO-based model achieves the second highest correct rate of 69.70%, better than YOLO v1 (65.50%) and is almost the same as Fast R-CNN (71.60%), while Fast R-CNN is much more complex and difficult to train than our model.

Besides, compared with the performance of another single-stage rotated object detection method, R^{3}Det, which reaches an accuracy of 86.43% on the ICDAR2015 dataset [3], the YOLO-based model achieves relatively better results. Considering the speed of R^{3}Det is only 13 FPS, our model will have an advantage over R^{3}Det in real-time processing or edge-computing devices.

Bounding boxes are highlighted in green ellipse and confidences are indicted in red text.

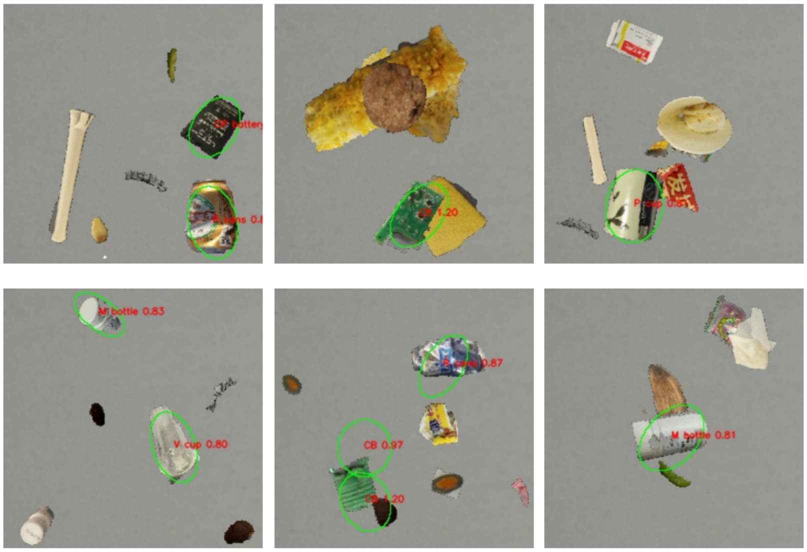

Figure 4: Garbage Detection and Classification Results

Table 2: Performance of the Proposed YOLO-based Model and Other Existing Models

Levels	*The Proposed Model*	YOLO	Fast R-CNN
Correct	69.70%	65.50%	71.60%
Localization	22.10%	19.00%	8.60%
Similar	0.30%	6.75%	4.30%
Other	1.60%	4.00%	1.90%
Background	6.30%	4.75%	13.60%

Table 3: Speeds of the Proposed YOLO-based Model and Other Existing Models

Model	Speed / FPS	Model	Speed / FPS
RCNN	2	RCI & RC2	Slow
RRPN	3.5	RPN	Slow
RetinaNet-H	14	RRD	Slow
RetinaNet-R	10	ROI-Transformer	6
R^3Det	10	YOLO	45
The Proposed Model	60	. . .	. . .

Table 4: Model Sizes of the Proposed YOLO-based Model and Other Existing Models

Model	FLOPs / B	#Parameters / M
SSD	34.36	34.30
YOLO v2	17.50	67.43
VGG16	15.30	138
The Proposed Model	6.40	32.1

5.3 Speed

Among real-time detectors, our model achieves the fastest speed among the existing models, as shown in Table 3

5.4 Model Size

Our model is smaller with regard to the model size measured by the number of parameters. Smaller model size makes our model more

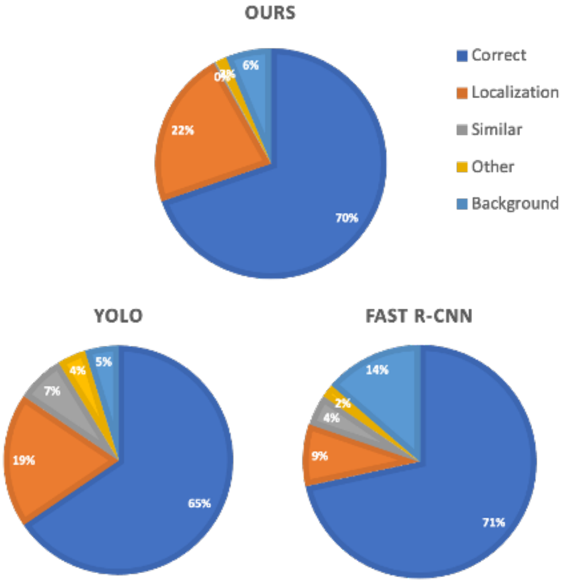

Figure 5: Performance of the Proposed YOLO-based Model and Other Existing Models

suitable for portable deployments and edge-computing devices. Another point worth mentioning is that our model also requires less calculation measured by Floating Point Operations Per Second (FLOPs) compared with other existing models.

5.5 Comparison between the Models with AVE and without VAE

During the experiment, we assumed that the VAE in our proposed model can improve the accuracy of the proposed neural network for regression tasks and mitigate the over-fitting problem. To validate these assumptions, we trained two models, one with VAE, and the other without VAE but has exactly the same structure as the previous one.

Fig 6. shows the results of the difference in orientation with respect to the number of epochs. The blue-dotted line represents the predicted difference in orientation of our proposed model without VAE, and the orange-dotted line represents the predicted difference in orientation of the model with VAE. The red solid line is generated by polynomial fitting of the blue-dotted line (without VAE), and the green solid line is generated by polynomial fitting of the orange-dotted line (with VAE).

From Fig. 6, our model with VAE performs better than that without VAE. The average difference in orientation is 20{°} for the model with VAE and 22{°} for the model without VAE. We can also observe an over-fitting tendency with regard to the model without VAE, as adding a VAE to the proposed model is able to effectively mitigate its over-fitting problem. Besides, during our experiments, after training for more than 250 epochs, the YOLO-based model with VAE reached a correct rate of 69.70%, but without VAE it only obtained a correct rate of 47.10%.

6 CONCLUSION

In this study, we proposed a YOLO-based neural network model with VAE for intelligent garbage detection and classification. Compared with existing models such as YOLO v1 and Fast R-CNN, the proposed model can achieve a relatively high correct rate of 69.70% with a faster speed and a smaller model size, and is also more robust

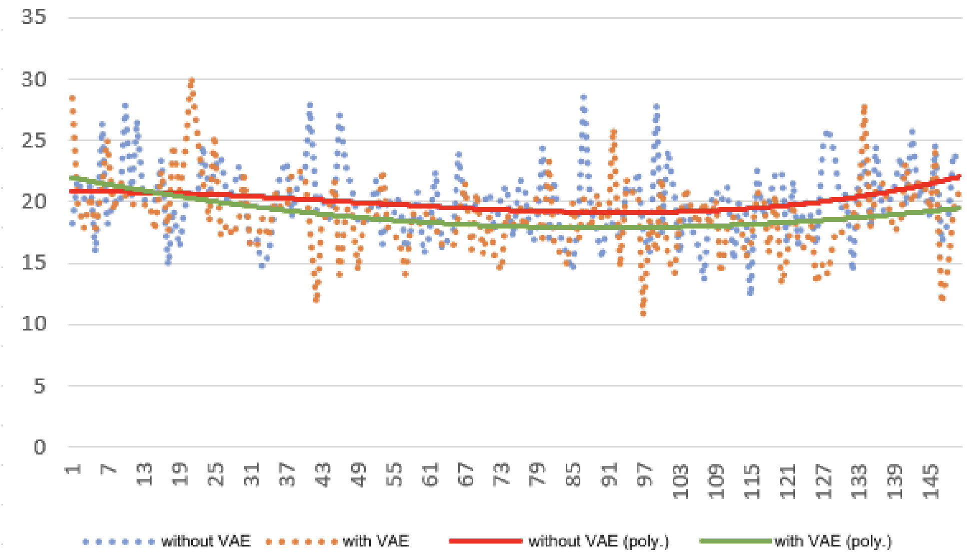

Figure 6: Performance of Models with VAE and without VAE

against over-fitting—these factors make our proposed model a good choice for portable edge-computing devices. Despite the advantages of this model, there still exist some limitations. For example, it is not sensitive to objects that are close to each other, because each predictor only predicts one bounding box for objects centering at a small grid. A possible solution to this problem under controllable industrial environment is to use robotic arms to separate piles of garbage by some mechanism. Besides, the accuracy of the model has a large space to be improved, since it is rather shallow at present. In the future, we can potentially improve its accuracy by using a deeper and more complex base net.

REFERENCES

[1] "The Growing Global Landfill Crisis", https://steelysdrinkware.com/growing-global-landfill-crisis/, Accessed 25 August 2020.

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, (2016), pp. 779-788.

[3] X. Yang *et al.*, "R^{3}Det: Refined Single-Stage Detector with Feature Refinement for Rotating Object", *arXiv preprint arXiv:1908.05612*, (2019).

[4] G. White *et al.*, "WasteNet: Waste Classification at the Edge for Smart Bins", *arXiv preprint at arXiv:2006.05873*, (2020).

[5] G. Thung and M. Yang, "Classification of Trash for Recyclability Status", (2016).

[6] O. Adedeji and Z. Wang, "Intelligent Waste Classification System Using Deep Learning Convolutional Neural Network", *The 2 nd International Conference on Sustainable Materials Processing and Manufacturing*, Sun City, South Africa, (2019), pp. 607-612.

[7] M. U. Sohag and A. K. Podder, "Smart garbage management system for a sustainable urban life: An IoT based application", *Internet of Things*, (2020).

[8] "2020 Haihua AI Challenge: Waste Sorting Task 1", https://www.biendata.xyz/competition/haihua_wastesorting_task1/, Accessed 25 August 2020.

[9] J. Redmon and A. Farhadi, "YOLO v3: An Incremental Improvement", *arXiv preprint arXiv:1804.02767*, (2018).

[10] S. Goswami, "Reflections on Non-maximum Suppression (NMS)", https://medium.com/@whatdhack/reflections-on-non-maximum-suppression-nms-d2fce-148ef0a, Accessed 25 August 2020.

[11] Q. Zhao *et al.*, "Variational AutoEncoder For Regression: Application to Brain Aging Analysis", *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, Shenzhen, China, (2019), pp. 823-831.

[12] I. Higgins *et al.*, "beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework", *International Conference on Learning Representations*, Toulon, France, (2017).

[13] Y. Yoo *et al.*, "Variational Autoencoded Regression: High Dimensional Regression of Visual Data on Complex Manifold", *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, (2017), pp. 2943-2952.