Deep Learning Approaches for Dynamic Object Understanding and Defect Detection

Yuan-Tsung Chang, W. K. T. M. Gunarathne, Timothy K. Shih

Department of Computer Science and Information Engineering, National Central University, Taiwan
{harry.500net, tharanga.gunarathne, timothykshih}@gmail.com

Abstract

Industrial product defect detection has been known for a while to make sure the released products meet the expected requirements. Earlier, product defect detection was commonly done manually by humans; they have detected whether the products consist of defects or not by using their human senses based on the standard. In this industrial era, product defect detection is expected to be faster and more accurate, while humans could be exhausted and become slower and less reliable. Deep learning technology is very famous in the field of image processing, such as image classification, object detection, object tracking, and of course the defect detection. In this study, we propose a novel automated solution system to identify the good and defective products on a production line using deep learning technology. In the experiment, we have compared several algorithms of defect detections using a data set, which comprises 20 categories of objects and 50 images in each category. The experimental results demonstrated that the proposed system had produced effective results within a short time.

Keywords: Deep learning, Image processing, Defect detection, CNN

1 Introduction

The manufacturing process takes place in the production system. Industrial product defect detection has been known for a while to make sure the released products meet the expected requirements. Earlier, product defect detection was performed manually by humans. However, such detection is very time consuming, inefficient, and can contribute to a serious limitation of the production capacity [1].

Internet is not a strange term since decades ago. It has been utilized all over the world as it is almost like a primary human need nowadays. It is also implemented in manufacturing sectors. A Cloud-ICT Convergence Service Architecture to construct an open and elastic digital information ecosystem is proposed by [2]. A method to model and analyze adaptive resource management for cloud applications that can be implemented in industry sector is also proposed by [3].

An automated metallic object defect detection using convolutional neural networks has been proposed by [4], and it performs well when detecting clear defects such as damage spots, dusts, and fibers but does not work well with vague defects and low-contrast scratches such as glues spots and scratches. The similar work has also been implemented by [5] to detect brain tumor, but it tends to miss the defect detection. Defect detection using improved Otsu method was also proposed by [6], but it easily over-detects which results in a large amount of background noise also being segmented [16].

Various algorithms are available to identify an object and detect the parameters, such as the size and appearance of its edges using image recognition technology. One of the examples is Convolutional Neural Network (CNN). CNN is a special type of multi-layer neural network inspired by the mechanism of the optical system of living creatures [7].

In the past, image recognition technology needed more time for computation and producing its result. Moreover, in the applications on dynamic motion detection, plenty of images were calculated. Therefore, applying deep learning is one of the solutions to reduce the time on calculation and to detect dynamic objects fast with high accuracy. Previously, the high accuracy and precision were challenging to achieve. To get results with high accuracy, some reinforcements of detection types of equipment were used: infrared devices, radar, and microscope-level AOI types of equipment [15].

The algorithms used in this paper are Yolo V3, SSD, R-CNN, and Mask R-CNN. Several categories of object defect, such as irregular shape, dent, scratch, damage, spot, and dirt. The algorithms attempt to identify the defects on an object in the images. Some devices are used to help the algorithms to identify the defects as a whole system. Therefore, high accuracy identification result is expected. We conducted the experiments using several algorithms of image processing to detect object defects. Besides, we used the CNN database as the image data set, which consists of 20 categories of objects. Each category includes 50

*Corresponding Author: Timothy K. Shih; E-mail: timothykshih@gmail.com
DOI: 10.3966/160792642020052103015

images, and the total dataset consists of 1000 images.

The paper is organized as follows: Section 2 presents related studies about different types of detectors and related techniques. Section 3 introduces the implementation of proposed model. Section 4 describes experiments and results. Finally, the section 6 describes the discussion and Conclusion.

2 Related Studies

2.1 YOLO (You Only Look Once)

Current detection systems re-planning classifiers to perform detection. In order to detect an object, these systems use a classifier and evaluate it at various locations and scales in a test image. Systems like deformable parts model (DPM) use a sliding window approach where the classifier could run at evenly spaced locations over the entire image [8]. YOLO reframes object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. In this work, we use YOLO to predict what objects are present and what they are.

In YOLO, a single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance. This unified model has several benefits over traditional methods of object detection. YOLO is a high-speed algorithm that processes an image in real-time at 45 frames per second. Another version of YOLO, Fast YOLO, processes 155 frames per second while still achieves double the mAP of other real-time detectors. Figure 1 shows how the YOLO detection system works.

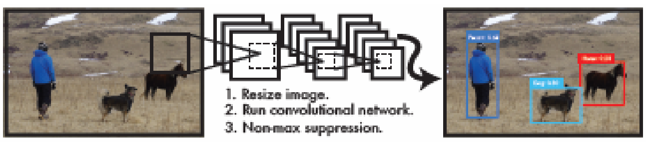

Figure 1. Representation of YOLO detection system. YOLO resizes the input image to 448 × 448. It runs a single convolutional network on the image and thresholds the resulting detections by the model's confidence

2.2 SSD (Single Shot Multibox Detector)

SSD is an algorithm for detecting objects in images using a single deep neural network. SSD discretizes the output spaces of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location [9].

When predicting defects, the network generates confidence scores for the existence of each object in the categories in each default box and makes adjustments to the box to match the object better. Also, the network combines predictions from several feature maps of different resolutions to comply with objects of various sizes.

SSD could be more simple relative to algorithms which require object proposals because it eliminates proposal generation and subsequent pixel or feature resampling stages and encapsulates the whole computation in a single network.

SSD results exhibit in significant improvement in speed for high-accuracy detection (59 FPS with mAP 74.3% on VOC2007). The improvement in speed is caused by eliminating the bounding box proposal and the subsequent pixel or feature resampling stage. SSD also improves accuracy compared to the other algorithms, which do a similar step by using a small convolutional filter to detect object categories and offset in bounding box locations using separate filters for other aspect ratio detections. Then apply those filters to several feature maps from the next stages of a network to perform prediction at multiple scales [9].

2.3 R-CNN

The deep convolutional neural network (CNN) has recently been achieved remarkable results in various fields. Especially in the area of visual recognition category in the large-scale visual recognition challenge. The main idea of CNN is that multiple convolutional layers, associated weights, and pooling layers are now included in the features proposed in the computer vision literature.

R-CNN is a simple and scalable detection algorithm which combines two key insights. First, we apply high-capacity convolutional neural networks (CNNs) to bottom up region proposals to localize and segment objects. Second, when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost [10]. Therefore, R-CNN combines region proposals with CNN features. R-CNN system overview is shown in Figure 2.

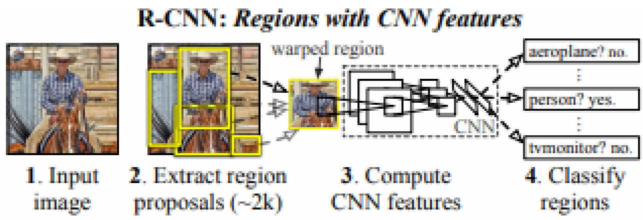

Figure 2. Representation of R-CNN system. It (1) takes an input images, (2) extracts about 2000 bottom-up region proposals, (3) computes features for each proposal using a large CNN, then (4) classifies each region using class-specific linear SVMs

R-CNN has other two upgraded versions: Fast R-CNN and Faster R-CNN. Compared to R-CNN, Fast R-CNN has improved in three aspects: (1) The speed of the test is faster because it solves the problem where R-CNN overlaps with a large number of frames proposals within images which results in a considerable amount of redundancy in the extraction feature operations. (2) The speed of training is also faster. (3) It requires smaller storage compared to the R-CNN, which requires a large number of features as training samples.

Faster R-CNN has several upgrades, such as R-CNN and Fast R-CNN. Its most significant difference from the previous two is the four steps required for target detection, which are region proposal generation, feature extraction, classification, and regression. All those four steps are done in the deep neural network and run on GPU, which significantly improves the efficiency of the operation. Figure 3 shows the structure comparison of R-CNN, Fast R-CNN, and Faster R-CNN.

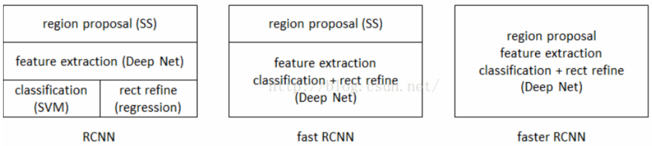

Figure 3. The structure comparison representation between R-CNN, Fast R-CNN, and Faster R-CNN

Faster R-CNN uses CNN to extract image features, then uses region proposal network (RPN) to extract ROI. Next, it uses ROI pooling to turn these ROIs into fixed size, then feeds the fully connected layer for bounding box regression and classification prediction.

2.4 Mask R-CNN

Mask R-CNN (MRCNN for short) is based on the R-CNN series, feature-based pyramid network (FPN), and fast causal inference (FCI). In general, it often improves Faster R-CNN. The main idea of Mask R-CNN is a Faster R-CNN whose each region proposal has two outputs, the category tag and the offset of the box. Then, Mask R-CNN adds another branch based on Faster R-CNN and adds another output, the object mask [11].

Mask R-CNN is an image processing algorithm which is conceptually simple, flexible, and general framework for object instance segmentation. It should efficiently detect objects in an image while simultaneously generating a high-quality segmentation mask for each instance. It can also be used to estimate human moves or poses in the same framework.

Mask R-CNN extends Faster R-CNN by adding a branch for predicting an object mask in parallel with the existing branch for bounding box recognition, while it is simple to train and only adds a small overhead compared to Faster R-CNN.

3 Implementation of Proposed System

In this paper, we conducted experiments using several algorithms to detect defects on objects based on object categories. The goal of this work is to implement a system which utilizes the best algorithms in terms of prediction accuracy and time consumption during the training and testing phase along with our infrastructure as a whole system. We applied several open-source frameworks, such as TensorFlow, Keras, and Pytorch, to support the experiments. Some algorithms were implemented by utilizing more than one frameworks. Besides, we conducted experiments on each of them to produce results.

The algorithms we used in this paper include YOLO V3, SSD300, SSD512, Faster R-CNN, and Mask R-CNN. The main concern of the result comparison is the accuracy of the prediction against the images in the data set.

We used several auxiliary devices to conduct the experiments, such as CCD lens, lighting sources, PCs, and graphic card or GPU. CCD lens was used to take images of the objects, while lighting sources were supposed to support the image taking to make sure that the objects are illuminated properly. Therefore, the images produced are expected to be more acceptable as data set. Also, graphics cards or GPUs were installed into the PCs. They were supposed to help the computation to be more productive and to make the computation time shorter. The setting of the devices used is shown in Figure 4. To make sure the experiments run correctly, we prepared an area with a size of about 200cm × 200cm.

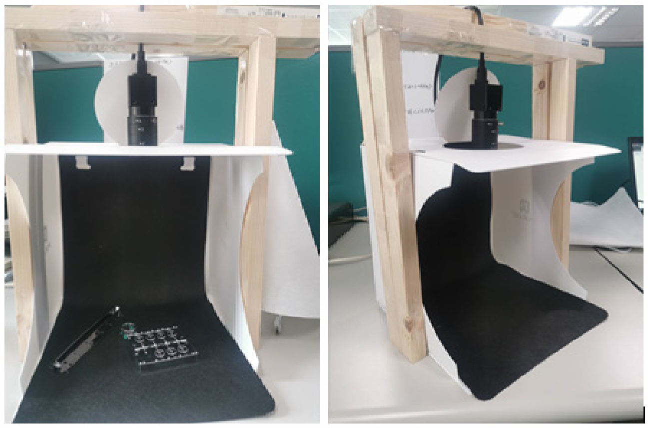

Figure 4. The setting of the devices used to conduct the experiments. The CCD lens is placed in a fixed position with enough lightings above the objects whose images are about to be taken with a pad below them

3.1 Equipment Installation

The object aspects concerned include light reflections, arcs, surfaces, a field of view, contours, and other common geometric shapes. The defects expected to be detected are scratches, shavings, white spots, cracks, and uneven paint.

To ensure the experiments run well, we selected the CCD lens and other hardware correctly concerning the quality. Then, we conducted lighting and optical test to make sure that the objects were illuminated appropriately to produce clear images. The next things to be tested before the experiments were the AOI image processing algorithms we used.

3.2 Image Data Collection

Next, we collected image data by taking pictures using the pieces of equipment prepared. CCD lens captures pictures of the object of each categories. The image format is .jpeg, and the image size is 1920 × 1080 pixels. There are 20 object categories and 50 images for each categories. Therefore, there is a total of 1000 images for the data set. In the experiments, we annotated the defects on object images using LabelImg application. We divided the image data set into training and testing set along with the XML annotations produced by LabelImg by a ratio of 8:2. Image data category examples are shown in Figure 5 and Figure 6.

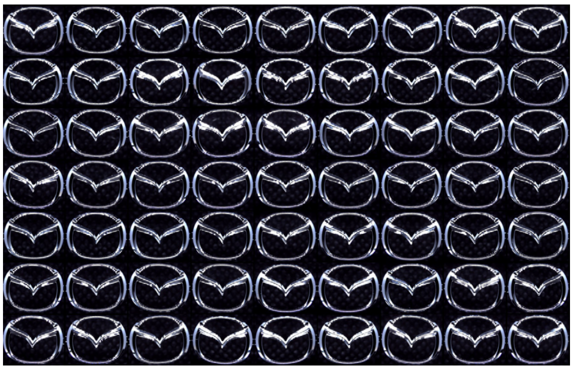

Figure 5. A sample of object image data used in our experiments called category 1

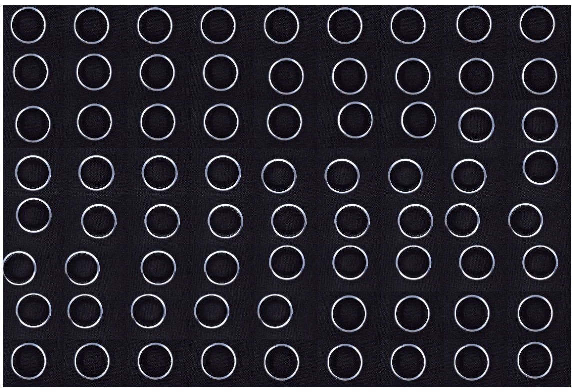

Figure 6. Another sample of object image data used in our experiments called category 2

3.3 Defect Detection Experiments

After collecting the data, we conducted the testing using the AI deep learning framework and algorithms we used to make sure there is no fault or bug inside them. Then, we placed settings and adjustments to the AOI and AI, so they could work together to achieve our goal.

We defined the scope and threshold of the defect detection according to the customized requirements of each object to meet the requirements of actual quality. The scope and threshold were the parameters for the classification to decide which class an object in the image is. Figure 7 shows several examples of defects on object in the images. There were 4 categories of object defect, they were shavings, uneven paint, scratch, and damage. Those defects can make an item as a fail one, so that it cannot be released or sold to the public.

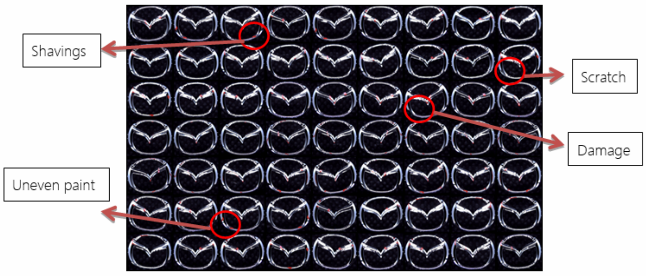

Figure 7. Examples of defects on objects

We started the experiments once the required settings were confirmed. Before doing training, we did data augmentation. The purpose of data augmentation is to acquire good-quality data and prevent uneven class balance within the datasets [12-13]. Next, we created a model using the algorithms defined above with the training data set as the input data. After getting models from the training process, we performed testing on the models against the testing data. Each object image was classified using all the algorithms and frameworks mentioned. They were classified into one out of two classes, either OK or NG. The OK class indicates that the object in the image is qualified as a good one. On the contrary, the NG class indicates that the object in the image is not qualified and does not meet the requirements. The prediction results of each algorithms were then compared against the ground truth. Finally, we obtained the accuracy results of each algorithms on each image categories and recorded them.

4 Experiments and Results

Deep learning is a branch of machine learning based on a set of algorithms which model high-level abstractions or pattern in data using multiple processing layers with complex structures or multiple non-linear transformations [11-16]. It is an algorithm that uses artificial neural networks as a framework to characterize and learn data. Therefore, most deep learning methods use a neural network architecture, which is why deep learning models are often referred to as deep neural networks [17-18].

The image processing problem has been long one of the most discussed areas in deep learning or the machine learning field. When it comes to digital image processing, there are many supporting algorithms to be used with input data and we can avoid some processing problems like noise creation and signal distortion at some point in signal processing as an advantage over analog image processing [19]. As technology grows, the need for automation in many segments also rises. Not only processing speed, but accuracy has also been the most concerned parameter. An image processing system should be able to detect or predict an object with high accuracy within the expected processing time.

In the industrial sector in this globalization era, a rapid decision in production is necessary. Therefore, product defect detection could help in the quality assurance sector to decide whether a product meets the requirements or standards or not. The automated defect detection system is required to meet the needs of quick decision and accuracy and to make sure the quality assurance runs swiftly.

In our experiments, preparing the sample object images was a crucial step. Even the conditioning or setting of the image taking has to be concerned to get the expected quality of the data set. In order to make sure the experiments give the most optimum and objective results, the algorithms and frameworks used in the model were tested before conducting the real experiments, confirming there was no error or bug inside them.

4.1 YOLO (You Only Look Once)

YOLO unifies the separate components of object detection into a single neural network. The network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes for an image simultaneously. It means that the network reasons globally about the full image and all the objects in the image. YOLO designs make end-to-end training and real-time speeds possible while maintaining high precision [8].

YOLO's final layer predicts class probabilities and bounding box coordinates. Besides, YOLO uses a linear activation function for the final layer, and all other layers use the following modified linear activation function:

$$\phi(x) = \begin{cases} x, & if\ x > 0 \\ 0.1x, & otherwise \end{cases} \quad (1)$$

YOLO predicts several bounding boxes for each grid cell. During the training period, only one bounding box predictor was assigned for one object. We assigned one predictor to predict objects based on which prediction has the highest current IOU with the ground truth. This matter causes specialization between the predictors. Each predictor becomes specialized at predicting specific sizes, aspect ratios, or object's classes which makes it better to detect particular objects and improve the prediction accuracy. Figure 8 shows the architecture of YOLO.

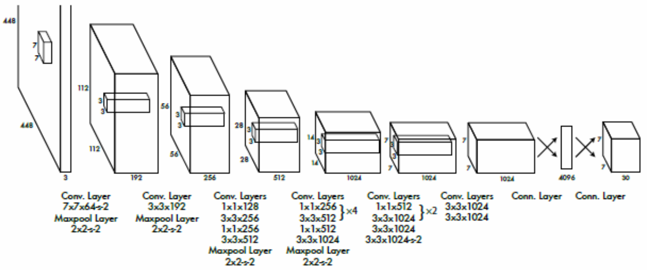

Figure 8. The architecture of YOLO. The detection network has 24 convolutional layers followed by 2 fully connected layers. It pretrains the convolutional layers on the ImageNet classification task at half the resolution (224 × 224) then double the resolution for detection

4.2 SSD (Single Shot Multibox Detector)

SSD is based on a feed-forward convolutional network that makes fixed-size bounding boxes and scores for the existence of an object in those boxes with a non-maximum step to get the final detections [9].

In the training stage, the significant difference between SSD and other algorithms in object detection that uses region proposals is that ground truth was assigned to specific outputs. If the assignment is determined, it applies loss function and backpropagation end-to-end. During the training, it is necessary to determine which default boxes correspond to ground truth and train the network accordingly. For each ground truth box, it selects from the default boxes of various locations, aspect ratio, and scale. Then, it compares every ground truth box to the default box with the best Jaccard overlap, which is higher than a threshold (0.5). This process lets the network to predict high scores for several overlapping default boxes with the ground truth boxes. So, it does not only pick the one with maximum overlap. SSD Framework is shown in Figure 9.

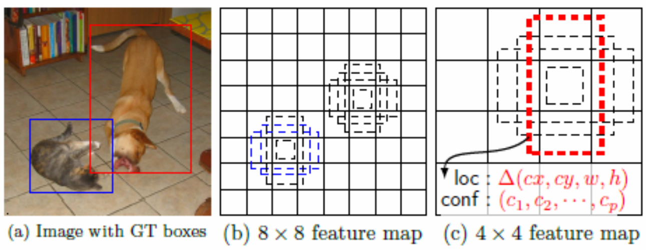

Figure 9. A representation of SSD framework

4.3 R-CNN

R-CNN consists of three modules: The first one produces category-independent region proposals; these proposals determine the candidate detections available to the detector. The second one is a big convolutional neural network which extracts a fixed-length feature vector from every region. The third one is a collection of linear SVMs, which is specific to classes.

4.4 Mask R-CNN

Like in Fast R-CNN, a region proposal was considered as positive if it has IoU of a minimum of 0.5. Otherwise, it is considered negative. The mask loss L_{mask} is determined only on positive region proposals. The mask target is the intersection between a region proposal and its corresponding ground-truth mask.

4.5 Result Comparison

We recorded the experiment results with several parameters. The parameters were training time in hours, testing time in seconds, and accuracy. We compared the results of each algorithms and frameworks. The experiment results are shown in Figure 10, Figure 11, and Figure 12.

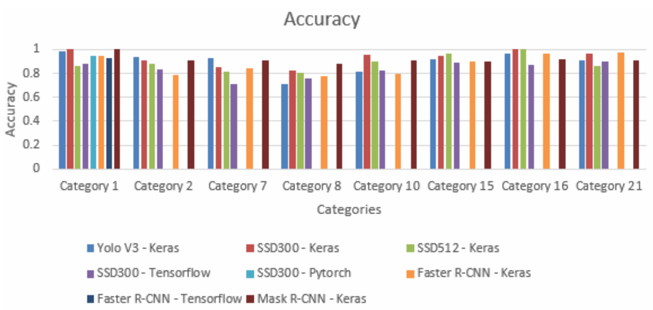

Figure 10. Performance comparison between the algorithms and frameworks on each categories in terms of accuracy

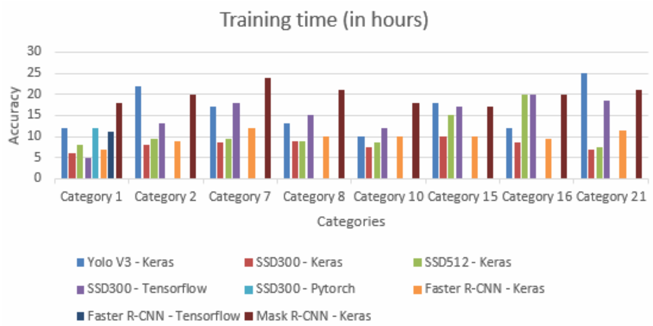

Figure 11. Performance comparison between the algorithms and frameworks on each categories in terms of training time (in hours)

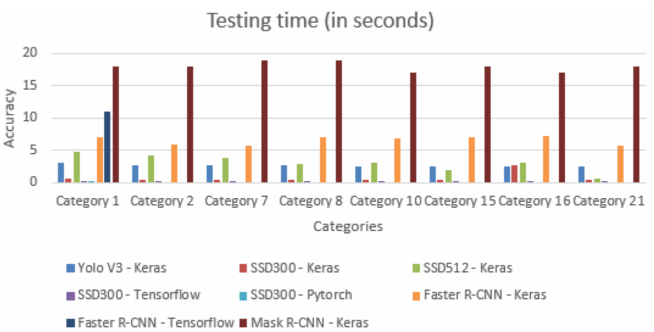

Figure 12. Performance comparison between the algorithms and frameworks on each categories in terms of testing time (in seconds)

Based on the results, looking at the accuracy average, we can see that SSD300 algorithms with Pytorch framework are the best ones. However, we conducted the experiments using the algorithms and framework for only once with one category. Therefore, it is not valid to conclude based on this stage. SSD300 with Keras as the framework gives the best accuracy at average. It predicts category 1 and 16 correctly but does inadequate for category 7 and 8. Therefore, SSD300 in Keras is the best at average but is not consistent. In terms of consistency, Mask R-CNN in Keras could be considered the most consistent one.

Training time is also an important parameter to evaluate the effectiveness of an algorithm. Based on the results above, SSD300 in Keras has the lowest average training time, while Mask R-CNN in Keras results in the longest one. It has proven that SSD300 could outperform Faster R-CNN in both speed and accuracy using the VOC data set. The training speed rate of SSD300 is most likely because it eliminates region proposal generation and subsequent pixel or feature resampling and encloses the whole computation in a single network. We also measure the performance of algorithms by how fast the model can detect objects. Our experiment results in testing time show that SSD300 in Pytorch produces the best results. However, it is tested only once using one category. Both SSD300 in Keras and Tensorflow spend the lowest time in the testing phase. If we combine the result with the one in Pytorch, we can completely conclude that SSD300 is the fastest algorithm in the testing phase.

SSD300 has not only the best accuracy but also proves that it has the fastest time for the training and testing phase. Even though its accuracy is not consistent over the categories, it is still the best one at average. Faster R-CNN is an upgraded version of CNN that has shorter processing time, but SSD300 still outperforms it in terms of both accuracy and either training and testing time. SSD300 can even be modified by changing the base network with a faster one to speed up its training and testing time.

5 Discussion and Conclusion

Defect detection of the product image was discussed in this paper. Due to the fast development of technology, image processing fields become more accessible. Nowadays, there are speedier hardware available that can be used to accelerate computations. Those hardware may help experiments to be more effective in terms of time. In the past, the graphics processing unit (GPU) was only used to support the display on the monitor. However, these days, GPU may also be used to help in mathematical computation and could be faster than a regular processor. Image processing, including defect detection, could be called a field with heavy computation, which needs excellent infrastructures to be more effective in terms of time.

Therefore, we used four computers and GPUs to do our experiments.

In the industrial field, quick defect detection on products is necessary because of the rapidity of the production itself. Before being released to the public, products must be confirmed to be qualified and meet the requirements and standards. The defective ones must be separated from the qualified one. Beside quickness, automation is also a need, because manual defect detection by human has drawbacks as humans might get tired and make the wrong decision. Therefore, automatic defect detection using image processing technology is a solution.

The contribution of this paper is to build a system with the best algorithm in the field of image processing based on our experiments by comparing multiple existing algorithms, along with our infrastructures, in order to implement an automated object's defect detection in real time. The algorithms we compared in this paper include You Only Look Once (YOLO), Single Shot Multibox Detector (SSD), Faster R-CNN, and Mask R-CNN. Those algorithms have been discussed in the world of object detection. They have shown excellent results in terms of accuracy and training time. Thus, we conducted experiments to compare them using our own produced data set. The parameters compared in this paper are accuracy, training time (in hours), and testing time (in seconds). We wanted to identify which algorithms have the fastest time in the training and testing phase and which one predicts defects on objects the best in terms of accuracy.

Based on our experiment result, SSD300 shows excellent results. It outperforms all algorithms in accuracy and either training and testing time. Even though it does not confirm consistent accuracy over object categories, it is still an excellent algorithm for outperforming the others in those three parameters. We also compared several similar algorithms but in different frameworks, and they do not show significant differences against the accuracy, but the visible difference in training and testing time. Most likely, it is caused by code implementation only, but how the algorithm works is still the same.

References

[1] D. Tabernik, S. Šela, J. Skvarč, D. Skočaj, Segmentation-Based Deep-Learning Approach for Surface-Defect Detection, *Journal of Intelligent Manufacturing*, Vol. 31, No. 3, pp. 759-776, March, 2020.

[2] L. J. Tan, Q. Wu, C. M. Wu, An Elastic Reconstruction of 5G Network Oriented Cloud-ICT Convergence Service, *Journal of Internet Technology*, Vol. 18, No. 5, pp. 1203-1214, September, 2017.

[3] L. Chen, G. Fan, H. Yu, Formally Modeling and Analyzing the Adaptive Resource Management for Cloud Applications, *Journal of Internet Technology*, Vol. 18, No. 5, pp. 1003-1015, September, 2017.

[4] T. Kalaiselvi, P. Nagaraja, A Rapid Automatic Brain Tumor Detection Method for MRI Images Using Modified Minimum Error Thresholding Technique, *International Journal of Imaging Systems and Technology*, Vol. 25, No. 1, pp. 77-85, March, 2015.

[5] X. Tao, D. Zhang, W. Ma, X. Liu, D. Xu, Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks, *Applied Sciences*, Vol. 8, No. 9, Article 1575, September, 2018.

[6] X. Yuan, L. S. Wu, Q. Peng, An Improved Otsu Method Using the Weighted Object Variance for Defect Detection, *Applied Surface Science*, Vol. 349, pp. 472-484, September, 2015.

[7] F. Sultana, A. Sufian, P. Dutta, Advancements in Image Classification Using Convolutional Neural Network, *Fourth International Conference on Research in Computational Intelligence and Communication Networks*, Kolkata, India, 2018, pp. 122-129.

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

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, A. C. Berg, SSD: Single Shot Multibox Detector, *European Conference on Computer Vision*, Amsterdam, The Netherlands, 2016, pp. 21-37.

[10] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *IEEE conference on computer vision and pattern recognition*, Columbus, Ohio, USA, 2014, pp. 580-587.

[11] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, *IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 2980-2988.

[12] A. Mikołajczyk, M. Grochowski, Data Augmentation for Improving Deep Learning in Image Classification Problem, *International Interdisciplinary PhD Workshop*, Swinoujście, Poland, 2018, pp. 117-122.

[13] L. Deng, D. Yu, Deep Learning: Methods and Applications, *Foundations and Trends in Signal Processing*, Now Publishers, Vol. 7, No. 3-4, pp. 197-387, June, 2014.

[14] Y. Bengio, Learning Deep Architectures for AI, *Foundations and Trends in Machine Learning*, Vol. 2, No. 1, pp. 1-127, January, 2009.

[15] Y. Bengio, A. Courville, P. Vincent, Representation Learning: A Review and New Perspectives, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 35, No. 8, pp. 1798-1828, August, 2013.

[16] J. Schmidhuber, Deep Learning in Neural Networks: An Overview, *Neural Networks*, Vol. 61, pp. 85-117, January, 2015.

[17] Y. LeCun, Y. Bengio, G. Hinton, Deep Learning, *Nature*, Vol. 521, pp. 436-444, May, 2015.

[18] I. Arel, D. C. Rose, T. P. Karnowski, Deep Machine

Learning- A New Frontier in Artificial Intelligence Research [Research Frontier], *IEEE Computational Intelligence Magazine*, Vol. 5, No. 4, pp. 13-18, November, 2010.

[19] J. Kuruvilla, D. Sukumaran, A. Sankar, S. P. Joy, A Review on Image Processing and Image Segmentation, *International Conference on Data Mining and Advanced Computing*, Ernakulam, India, 2016, pp. 198-203.

Biographies

Yuan-Tsung Chang is the CEO of a Computer System integration company in Taiwan and received his master's degree in Computer Science and Information Engineering at Tamkang University. He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering, National Central University, Taiwan. His research interests include machine learning, Image Processing, RFID signal Processing, System integration solutions and techniques.

W. K. T. M. Gunarathne received his B.Sc. in Computing Science and M.Sc. degree in E-Business from Staffordshire University, in the United Kingdom. He is currently a Ph.D. candidate in the Department of Computer Science and Information Engineering, National Central University, Taiwan. His research interests include e-learning, information retrieval and extraction, social data mining, and recommendation techniques.

Timothy K. Shih is a Distinguished Professor at the National Central University, Taiwan. He was the Dean of the College of Computer Science, Asia University, Taiwan and the Chairman of the CSIE Department at Tamkang University, Taiwan. Prof. Shih is a Fellow of the Institution of Engineering and Technology (IET). He was also the founding Chairman Emeritus of the IET Taipei Local Network. In addition, he is a senior member of ACM and a senior member of IEEE. (http://www.csie.ncu.edu.tw/~tshih/)