Leveraging Transdisciplinary Engineering in a Changing and Connected World
P. Koomsap et al. (Eds.)

doi:10.3233/ATDE230678

Integrating Deep Learning Models and Depth Cameras to Achieve Digital Transformation: A Case Study in Shoe Company

Li-Sheng YANG and Ming-Chuan CHIU
Department of Industrial Engineering and Industrial Management, National Tsing Hua University, Taiwan

Abstract. In today's fiercely competitive industrial environment, digital transformation and smart manufacturing have become important strategies for enhancing competitiveness. Digital transformation utilizes advanced technology and data analysis to make manufacturing processes more intelligent and automated, improve product quality, reduce production costs and time, and increase production efficiency. Smart manufacturing further applies machine learning, deep learning, and artificial intelligence to make the production process even more intelligent and automated. However, existing object detection models such as YOLO can only provide rectangular bounding boxes and cannot determine the actual rotation angle and inclination of objects, and lack discourse on hardware integration. Therefore, this study proposes a deep learning-based method framework that combines Yolov5 and Mask R-CNN to detect objects in real-time and calculate the object's center point coordinates, reference point coordinates for rotation direction, and inclination angle. This is integrated with a depth camera to obtain the distance between the robotic arm and the object, providing all the information required for the robotic arm to grasp the object. In simulated scenarios of stacking shoe insoles, the model proposed in this study achieved an accuracy of 97%. This technology can be applied in the factory production process, allowing robotic arms to accurately grasp objects from cluttered piles at the correct coordinates and angles, and perform sorting and assembly tasks. It can also help companies reduce costs and errors caused by human intervention, thereby enhancing their competitiveness.

Keywords. Robotic arm, Yolov5, Mask R-CNN, stacked objects, depth camera

Introduction

Many factories are gradually introducing automation technology in order to reduce production costs, minimize human errors, and increase output. However, in computer vision, using a robotic arm to grip objects is a classic problem, especially in practical industrial applications where stacking and irregularly arranged objects are common. For example, in certain processes that use injection molding machines to manufacture products, the products that are ejected and dropped onto the collection platform will be stacked at different angles, requiring additional sorting and arrangement processes. This not only increases production costs but also extends the entire production cycle time. To solve this problem, manufacturers can use two methods to extract targets from stacked

objects at the correct angle: one is to let personnel manually pick up objects, but with increasing work time, personnel are prone to fatigue, and accuracy will decrease. The other is to use computer vision to assist robotic arms in gripping objects. This method can make overall performance more stable and is relatively cheaper than human labor.

This study integrated the results of two object detection models, Yolov5 and Mask RCNN, to detect objects that can be successfully gripped in the case of stacked objects. It not only identifies their position, tilt angle in front, back, and left and right directions, but also calculates the clockwise rotation angle to achieve real-time detection.

1. Literature Review

1.1. Development of Object Detection

In the field of deep learning, object detection models can be divided into two-stage detection and one-stage detection. Two-stage detection represents models that focus on computational speed and pursue accuracy. On the other hand, one-stage detection represents models that aim to complete the detection process in one step and prioritize computational speed [1].

The most well-known model for one-stage detection is YOLO. YOLO treats the prediction of bounding boxes and object class recognition as a regression problem, using only one CNN to process the image and without the need for candidate region proposals. This allows YOLO to detect objects faster. Its calculation method involves dividing the input image into S*S grids, where each grid predicts B bounding box coordinates, confidence scores, and probabilities of object classes. Finally, the model outputs the best bounding boxes and object classes using non-maximum suppression (NMS). Table 1 shows the evolution of YOLO versions.

Table 1.、Comparison of joint detection methods.

	Advantages	**Disadvantages**
YOLOv1[2]	One of the advantages of the one-stage detection model is that it has fast detection speed, up to 45 FPS.	The recognition of nearby objects is poor, and each grid can only recognize one object.
YOLOv2 [3]	The input image is no longer restricted to a fixed size, and any input dimension can run throughout the entire network. The speed has been increased to 67 FPS.	The recognition of small and nearby objects is still poor, and the drawback of YOLOv1 cannot be solved.
YOLOv3 [4]	The main highlight is not in speed, but in the detection capability and quantity of small objects.	Performance drops significantly in the metric of map>0.5, and the final map is worse than RetinaNet.
YOLOv4 [5]	Efficiency, accuracy, and detection speed are all better than YOLOv3.	The model has a larger storage capacity, which makes it unsuitable for use on mobile devices.
YOLOv5 [6]	The model has a small storage capacity and boasts faster detection speed and excellent precision.	Translation: Slightly inferior to YOLOv4 in originality and precision.

The typical example of two-stage detection is the Region-Based Convolutional Neural Network (R-CNN) model proposed by Girshick in 2014 [7]. He et al. proposed Mask R-CNN based on Faster R-CNN [8]. Mask R-CNN can not only accurately detect various objects in the image but also draw masks based on the object's contour to achieve instance segmentation, which will be introduced in Section 3-2. The Mask R-CNN method has been widely used in various industries, such as Burke et al. [9] using Mask R-CNN to classify stars and galaxies in astronomical images, Hu et al. [10] applying it to identify lung regions in chest X-ray images, and Yang et al. [11] and Li et al. [12] using Mask R-CNN for object recognition in remote sensing photos. Jia et al. used Mask R-CNN to recognize overlapping apples in the forest to improve the accuracy of the automatic harvesting robot [13]. The evolution and comparison of two-stage detection models are shown in Table 2.

Table 2.、Comparison of joint detection methods.

	Advantages	**Disadvantages**
R-CNN [14]	Selective search 2000 possible regions.	Each region needs to be adjusted to the same size, which leads to slow computation.
SPP-net [15]	The pooling layer can take inputs of multiple scales and produce fixed-size output feature maps.	The pooling layer extracts feature maps at a slow speed.
Fast RCNN [16]	Combining R-CNN with a simplified SPP-net pooling layer, known as ROI pooling, can improve detection speed.	Corrected: Selective search to find all candidate boxes is still very time-consuming.
Faster RCNN [17]	Replacing selective search with RPN improved both precision and speed.	It can only output rectangular bounding boxes with no angle and their corresponding class.
Mask RCNN **Fehler! V erweisquelle konnte nicht gefunden werden.**[8]	Draw masks based on the object's contour to achieve instance segmentation.	It requires a larger dataset when performing multi-task training.

Two-stage and one-stage object detection models have their respective advantages in terms of detection accuracy and speed. Compared to one-stage detection, two-stage models require additional algorithms or neural networks to extract regions of interest (ROIs) in advance, from the earliest sliding window to selective search, and then to faster R-CNN with Region Proposal Network (RPN), which can improve the accuracy of bounding box location. However, the cost is that the detection speed is slower. On the other hand, one-stage detection solves the problem of object localization and classification simultaneously using a single neural network, making it faster than two-stage detection.

1.2. Smart manufacturing

Nowadays, the goal of the manufacturing industry is to enhance its competitiveness and ensure long-term growth by integrating cutting-edge information and communication technology. Smart manufacturing, considered the fourth industrial revolution, is seen as a new paradigm that redefines the operational model of manufacturing through intelligence-driven approaches [18]. Smart manufacturing utilizes modern information technologies such as the Internet of Things (IoT), artificial intelligence (AI), big data analytics, and machine learning to optimize the manufacturing process. It combines

production, machinery, products, and information technology to achieve more efficient, flexible, and sustainable production methods [19].

Deep learning has found numerous applications in the field of smart manufacturing. For example, in defect detection and quality control, the Mask R-CNN model can automatically detect defects in the wafer manufacturing process, reducing the need for manual inspection and improving product quality [20]. In predictive maintenance, deep neural networks (DNNs) can analyze sensor data from manufacturing equipment, predict equipment failures or maintenance needs, optimize maintenance plans, and reduce downtime and costs [21].

A robotic arm is a mechanical device that can mimic human arm movements, typically composed of a mechanical structure and a control system. It can be applied in various industrial, medical, and logistics applications [22]. Deep learning, as a subfield of machine learning, enables machines to automatically learn features from large amounts of data and make predictions or classifications. Object detection has seen significant development in the context of robotic arms, but there is limited literature describing subsequent applications that integrate hardware devices. For example, integrating object detection with depth cameras allows robotic arms to measure the distance between objects and the arm by utilizing depth information. The object's center position can be obtained through object detection for gripping purposes. Additionally, the depth camera can provide the normal vector of the object's plane, enabling the estimation of its tilt angle.

2. Research methods

The methodology of this research can be roughly divided into three stages. The first stage involves data preprocessing, which includes generating the training dataset and annotating the images. In the second stage, the images and annotated files are fed into the Yolov5 and Mask R-CNN models for training. The third stage involves integrating and analyzing the training results. By analyzing the results from Yolov5 and Mask R-CNN, the positions, tilt angles, and rotation angles of objects suitable for robotic arm grasping can be determined.

2.1. Data Preparation

In this research, the object patterns in the original images are individually removed, and then the Adobe Illustrator© software is used to randomly rotate and stack the objects sprayed onto the canvas. This process simulates a randomly arranged scene. A total of 1600 RGB images with a size of 1600*1200 pixels will be generated as the dataset. The labelme© software is used for manual annotation of the training data for Mask R-CNN. During the object annotation process, the complete and unobstructed objects are selected by drawing bounding boxes along their contours to extract the object masks. Roboflow is then used for manual annotation of the training data for Yolov5. While annotating the objects, efforts are made to place the center point of the bounding box within the detected object to facilitate subsequent operations. Finally, the annotated data for both Mask R-CNN and Yolov7 are used to train their respective models.

2.2. Model Training

We will divide section 3.2 into two parts. In section 3.2.1, we explain the architecture of the Yolov5 model. In section 3.2.2, we introduce the Mask RCNN model.

2.2.1 Yolov5

YOLO is a single-stage object detection method that uses a convolutional neural network architecture to determine the position and type of objects in an image, thereby improving recognition speed [23]. Therefore, YOLO is faster in detection than two-stage models. The network structure of YOLOv5 [23] consists of three parts: BackBone, Neck, and Output.

The BackBone is a convolutional neural network that aggregates different granularities of images and forms image features. The Neck layer fuses feature maps of different levels to obtain more contextual information, reduce information loss, and enhance the model's detection capability for objects of different scales. Through the image transformation convolutional features, the model predicts bounding boxes of different sizes [23]. After feature extraction through BottleneckCSP and repeated convolution, the Output layer establishes a convolutional kernel through Conv2D. This convolutional kernel convolves the layer input to generate the output tensor and obtain the predicted results of the target object.

2.2.2 Mask RCNN

The Mask RCNN is an extension of the Faster RCNN model for achieving instance segmentation. Instance segmentation is similar to general object detection, but the output result is a mask of the object instead of just a bounding box. Its purpose is to find the contour of the target object and distinguish between different individuals. The Mask RCNN adds a branch behind the Faster RCNN to predict the mask and also improves the original ROI pooling by introducing ROI align.

2.3. Model validation

To evaluate a model, we first use a confusion matrix to measure its performance on each task [24], which consists of counting the number of True positives (TP), True Negatives (TN), False positives (FP), and False negatives (FN). True positive indicates that the model successfully detects an object that exists in the image being detected; False positive means that the model detects an object where there is no object in the image; False negative indicates that the model fails to detect an object that exists in the image being detected; True negative means that the model correctly does not detect an object where there is no object in the image. As there are countless regions in the image being detected that do not contain objects of interest, True negative has little meaning in object detection tasks and its value is essentially infinite.

2.4. Analysis and Integration

This research utilizes Yolov5 to obtain the bounding box, center point, and depth information of objects. By applying singular value decomposition (SVD) to the nine sets of three-dimensional coordinates within the bounding box, the normal vector of the

object plane is calculated. The SVD approach decomposes the coordinate matrix, and the last column (or row) of the resulting matrix represents the normal vector. Based on the coordinates of the normal vector, the tilt angles in the left-right and up-down directions can be determined.

The analysis of Mask R-CNN results reveals that the generated masks consist of a matrix of True and False values (length * width * number of classes). Python's OpenCV package is then used to analyze each image. The preprocessing steps include blurring and grayscale conversion. The minimum bounding rectangle function is employed to draw the smallest rectangular box that fits the object contours. Additionally, the area of each object mask is calculated as a filtering criterion in the second round. If the mask area does not meet a predetermined standard (e.g., a fixed percentage of the complete object area), the aforementioned operations are not executed. Finally, the integration of Yolov5 and Mask R-CNN models is performed to determine the clockwise rotation angle of the objects.

Currently, the coordinates of the reference point and the center point of the object have been obtained using the aforementioned methods. The slope between the two points can be calculated, and then converted to radians. Arctan (inverse tangent) is commonly used to calculate the inverse tangent value of a given ratio. The input of the arctan function is a ratio (typically represented as a slope), and the output is the corresponding angle value. After obtaining the radians, they can be converted to degrees.

To confirm that the coordinates of the reference point and the center point belong to the same object, the two-dimensional matrix generated by Mask R-CNN, consisting of True and False values, is used for verification. If both the reference point coordinates and the center point coordinates fall within the generated mask by Mask R-CNN, it indicates that they belong to the same object, thereby determining the object's rotation angle. By integrating the above methods, complete information about the object can be obtained, including its three-dimensional coordinates, left-right tilt angle, up-down tilt angle, and rotation angle, totaling six values.

3. Case Study

The subject of this paper is a large-scale joint venture footwear manufacturing company, recognized globally for its shoe production and development technologies, and trusted by renowned international brands. In the production of insoles, the company utilizes injection molding machines, and the ejected insoles fall onto a collection platform. However, the improper collection and arrangement of the insoles can lead to line blockages and delays. Additional manpower and time are required to collect and organize the insoles, ensuring the operational efficiency of the production line. However, as working hours increase, employees not only become fatigued but also experience reduced efficiency in sorting the insoles. Therefore, the company aims to replace manual labor with robotic arms for the tasks of gripping and organizing the insoles, resulting in more stable overall performance and relatively lower costs compared to human labor, while also reducing the insole processing time.

In this chapter, the research will follow the steps described in Section 2 and apply them to the created dataset. In Section 3.1, the generation of training data will be explained, including image formats and annotation rules. In Section 3.2, the application of the custom dataset in YOLOv7 and Mask RCNN models and the generation of results will be introduced. The process of integrating the results from these two different models

will be described in Section 3.3. Finally, in Section 3.4, common object detection evaluation metrics will be used to assess the performance of the models.

3.1. Data Preparation

The case study involves a large Sino-foreign joint venture shoe manufacturing company that produces shoe insoles using injection molding. The resulting insoles are randomly placed on a receiving platform and overlap with each other. Obtaining relevant datasets for this scenario is difficult in real life, so this study used a few images provided by the manufacturer to remove the background of the insole pattern. Adobe Illustrator© software was then used to randomly rotate and spray the insole icons onto a canvas to simulate scattered insole images. A total of 1000 RGB images with a size of 1600*1200 pixels were generated as the dataset. Roboflow was used to label the entire insole and the head of the insole as the training dataset for Yolov5, followed by using labelme© software to label the masks. When labeling the insoles, only complete insoles in the images were selected.

3.2. Environment Setup for Training Models

3.2.1 Yolov5

In this study, we used Ultralytics' YOLOv5 to recognize shoe insoles and toe caps. The data for YOLOv5 was divided into 1000 training images and 30 test images. We used experimental design to optimize the parameters by adjusting the training steps (Epoch), learning rate, training batch size, and optimizer. We used the $L_9(3^4)$ orthogonal table for the experiment, and the accuracy was measured using the validation set. Finally, we used the Adam optimizer, set the learning rate to 10^-4, the batch size to 16, and the epoch to 300. The training time for one iteration was about 6 hours. During the training process, there was no significant fluctuation in the loss value of the training and validation sets, indicating that overfitting did not occur. At the end of the training process, the total loss value converged to about 0.002, which is less than the acceptable value of 0.05 (Kulkarni, Dhavalikar, and Bangar 2018), indicating that the model's prediction results were excellent. The accuracy of predicting shoe insoles was 98.2%, while the accuracy of predicting toe caps was 95%.

3.2.2 Mask RCNN

We used the open-source code of Mask RCNN on GitHub to implement the detection of insoles. The dataset used to train the Mask RCNN model consists of 272 training images and 28 test images, which are completely identical to the images used to train YOLOv5. After converting the annotated JSON file to the required files for the model (including the YAML file for storing labels and the PNG file for masks), we started training the model. The detection results of Mask RCNN generate pixel-level masks. After analyzing the code for the output results of Mask RCNN, we found that the object mask is composed of a True or False array (height * width * number of classes). Truly represents the masked area, while False represents the unmasked area.

3.3. Analysis and Integration of Results

After running Yolov5, we obtain the coordinates of the center points of the insole and the insole head. By using the results of both, we can calculate the clockwise rotation angle between the insole and the vertical line. We then use the Mask RCNN mask to determine if the object boxes of the insole and the insole head belong to the same insole. By performing SVD decomposition on the object box of the insole, we can obtain the normal vector of the box's plane to calculate the left-right and up-down tilt angles of the insole (Figure 1).

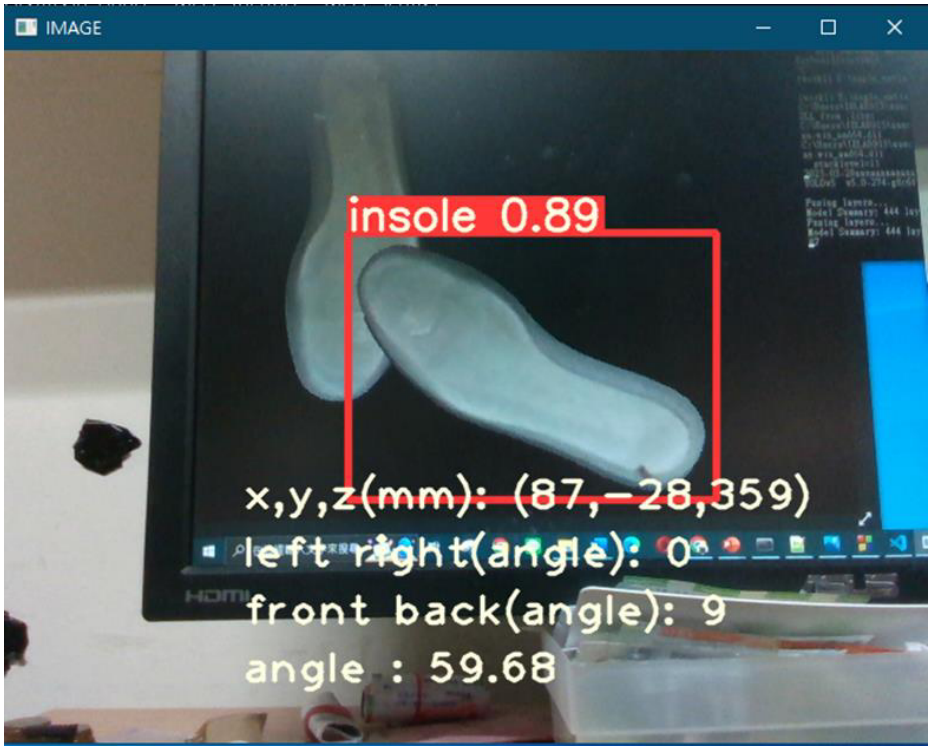

Figure 1. Object box of the insole.

3.4. Discussion

In this study, we combined Yolov5 and Mask RCNN models to overcome the limitations of previous research in practical applications. In real-world scenarios, objects are usually randomly arranged and stacked. In addition to incorporating angle information into the grasping conditions, it is also necessary to consider the issue of object tilting due to stacking. Moreover, real-time detection is often the best performance for application in robotic arms. This study solved several challenges in grasp detection by integrating two object detection models: (1) bounding boxes with angle information, (2) determining the actual angle of the object, (3) obtaining object tilt angle information, (4) integrating deep

learning into hardware devices to apply both functions and most importantly, (5) selecting suitable targets for grasping in overlapping situations.

4. Conclusion and Future Development

Due to the limitations of current object detection techniques in practical applications, this study integrates two deep learning models to address the real-world problem of stacking and irregular arrangement of objects. In the model validation tests, the proposed integrated model of YOLOv7 and Mask R-CNN achieved an accuracy rate of over 95% in identifying graspable objects. Since YOLO model is used as the foundation for overall detection and combined with depth cameras, real-time detection is achieved. The contributions of this study can be divided into academic and practical aspects. Academically, it integrates the results of two object detection models on the basis of real-time detection, improving both accuracy and efficiency. Additionally, it solves the problem of YOLO model's inability to calculate real angles. Finally, by calculating object angles using bounding boxes and integrating depth cameras, information about object tilt angles is obtained. In terms of practical applications, this study contributes in the following ways: firstly, the case study demonstrates high detection accuracy, which can replace manual labor and reduce labor costs. Secondly, the system enables real-time detection, shortening the grabbing operation time and improving production efficiency. Lastly, this study provides a system that offers real-time detection, distance measurement, determination of tilt angles, and rotation angles, reducing equipment expenses for businesses and possessing practical application value. Furthermore, the deep learning approach proposed in this study can be widely applied to different datasets, especially those with characteristics such as single-class objects, objects stacked on top of each other, directional properties, and random arrangement. Therefore, the proposed methodology has broad applicability in practical applications.

Future research directions include training the models using real-world datasets and increasing the number of samples in the dataset to improve detection accuracy. Additionally, incorporating more variations in images, such as lighting and shadows, can enhance the model's generalization capabilities. Furthermore, this study will explore techniques such as data augmentation and self-training to increase the quantity of training data. In real-world environments, overlapping of different objects is common, so efforts will be made to improve the model's ability to handle such scenarios.

References

[1] Z. Zou, Z. Shi, Y. Guo and J. Ye, Object detection in 20 years: A survey. *arXiv preprint* arXiv:1905.05055. 2019.

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

[3] J. Redmon and A. Farhadi. YOLO V2.0. *IEEE Conference on Computer Vision and Pattern Recognition 2017 (2017 Conference on Computer Vision and Pattern Recognition)*, 2017, April, pp. 187–213.

[4] J. Redmon and A. Farhadi, YOLOv3: An incremental improvement. *arXiv preprint* arXiv:1804.02767, 2018.

[5] A. Bochkovskiy, C.-Y. Wang and H.-Y. Mark Liao. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. 2020. view on: http://arxiv.org/abs/2004.10934.

[6] Ultralytics. YOLOv5. 2020, Retrieved from : https://github.com/ultralytics/YOLOv5

[7] R. Girshick, J. Donahue, T. Darrell and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.
[8] K. He, G. Gkioxari, P. Dollár and R. Girshick, Mask R-CNN. *2017 IEEE International Conference on Computer Vision (ICCV)*. 22-29 Oct. 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.
[9] C.J. Burke, P.D. Aleo, Y.-C. Chen, X. Liu, J.R. Peterson, G.H. Sembroski and J.Y. Lin, Deblending and classifying astronomical sources with Mask R-CNN deep learning. *Monthly Notices of the Royal Astronomical Society*, 2019, Vol. 490(3), pp. 3952-3965.
[10] Q. Hu, L.F.d.F. Souza, G.B. Holanda, S.S. Alves, F.H.d.S. Silva, T. Han and P.P. Rebouças Filho, An effective approach for CT lung segmentation using mask region-based convolutional neural networks. *Artificial Intelligence in Medicine*, 2020, 101792.
[11] F. Yang, T. Feng, G. Xu and Y. Chen, Applied method for water-body segmentation based on mask R-CNN. *Journal of Applied Remote Sensing*, 2020, Vol. 14(1), 014502.
[12] Y. Li, W. Xu, H. Chen, J. Jiang and X. Li, A Novel Framework Based on Mask R-CNN and Histogram Thresholding for Scalable Segmentation of New and Old Rural Buildings. *Remote Sensing*, 2021, Vol. 13(6), 1070.
[13] W. Jia, Y. Tian, R. Luo, Z. Zhang, J. Lian and Y. Zheng, Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot. *Computers and Electronics in Agriculture*, 2020, Vol. 172, 105380.
[14] R. Girshick, J. Donahue, T. Darrell and J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580-587.
[15] Y. Du, W. Wang and L. Wang, Hierarchical recurrent neural network for skeleton based action recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110-1118.
[16] R. Girshick, Fast R-CNN. *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.
[17] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'15)*. MIT Press, Cambridge, 2015, pp. 91–99.
[18] A. Biahmou, Towards agile enterprise rights management in engineering collaboration, *International Journal of Agile Systems and Management*, 2016, Vol. 9(4), pp. 302-325.
[19] H.S. Kang, J.Y. Lee, S. Choi, H. Kim, J.H. Park, J.Y. Son and S.D. Noh, Smart manufacturing: Past research, present findings, and future directions. *International journal of precision engineering and manufacturing-green technology*, 2016, Vol. 3, pp. 111-128.
[20] L. Li, B. Lei and C. Mao, Digital twin in smart manufacturing. *Journal of Industrial Information Integration*, 2022, Vol. 26, 100289.
[21] M.C. Chiu and T.M. Chen, Applying Data Augmentation and Mask R-CNN-Based Instance Segmentation Method for Mixed-Type Wafer Maps Defect Patterns Classification. *IEEE Transactions on Semiconductor Manufacturing*, 2021, Vol. 34(4), pp. 455-463.
[22] L. Ren, J. Cui, Y. Sun and X. Cheng, Multi-bearing remaining useful life collaborative prediction: A deep learning approach. *Journal of Manufacturing Systems*, 2017, Vol. 43, pp. 248-256.
[23] M.C. Chiu, H.Y. Tsai and J.E. Chiu, A novel directional object detection method for piled objects using a hybrid region-based convolutional neural network. *Advanced Engineering Informatics*, 2022, Vol. 51, 101448.
[24] B. Benjdira, Y. Bazi, A. Koubaa, K. Ouni Unsupervised Domain Adaptation Using Generative Adversarial Networks for Semantic Segmentation of Aerial Images. *Remote Sensing*, 2019, Vol. 11(11):1369. https://doi.org/10.3390/rs11111369.