Development of cost-effective IoT module-based pipe classification system for flexible manufacturing system of painting process of high-pressure pipe

Young-Jun Yoo1 · Ki-soo Cho2

Received: 16 July 2021 / Accepted: 26 November 2021 / Published online: 5 January 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
This paper researches an IoT module-based pipe classification system for a flexible manufacturing system that recognizes the size and length of pipes used in the painting process of high-pressure pipes. The proposed system is composed of an IoT module, USB camera, and edge TPU for pipe classification. The proposed system recognizes the type of pipe by three processes; object detection of the pipe, line detection algorithm of the three regions of interest, and pipe classification algorithm based on the line detection algorithm. Furthermore, the proposed system enables web-based real-time monitoring, providing convenience to workers and helping them make quick decisions. The IoT module interfaces with the painting robot and the sequence control that paints for each type of pipe is executed in the painting robot, allowing flexible manufacturing of the painting process.

Keywords Painting process · Pipe classification · Internet of things · Flexible manufacturing system

1 Introduction

Drilling refers to the operation of drilling a hole up to several tens of kilometers underground to produce oil and gas underground, and in this case, pipes and valves capable of withstanding high-pressure (15,000 psi) are required. High-pressure pipes have threaded, flanged, or union connectors at both ends and are widely used in oil drilling and production, splicing, and breaking and testing operations. Additionally, a high-pressure pipe with the straight tube with union connections at both ends is widely used to deliver high-pressure flow in break manifolds, junction manifolds, maintenance manifolds, and test manifolds.

A flexible manufacturing system is a system that automates production to manufacture various products with high productivity flexibly. In order to cope with the mix of different part types and the production volume adjusted according to changes in demand patterns, it is a productivity that simultaneously aims for flexibility and productivity of production that can process various parts types simultaneously. The goals pursued by the flexible production system are essentially flexibility, productivity, and reliability. In recent years, mass production and traditional manufacturing industries such as chemical and machinery, represented by fixed equipment, are also introduced for survival. Recently, the flexible production system is spreading as the industrial Internet, multi-functional robots, and big data technologies improve part identification ability, change processes quickly and significantly reduce setup costs.

Recently, numerous studies [1–9] have been conducted on flexible production. The paper [1] presents a general architecture of digital-twin visualization for flexible manufacturing systems (FMS) and for developing virtual digital-twin scenes architecture. Furthermore, the design of non-centralized control architectures is proposed [2] to improve their energy efficiency and promote flexibility. Also, seamless Human–Computer–Machine Interaction architecture is presented [3] to support the operator’s supervision activity.
with flexible manufacturing systems. Extended coloured stochastic Petri nets [4] are utilized to build a model for FMS with three machines for flexible production. Two heuristic algorithms (pre-determined fixed-job sequencing rules and genetic algorithm-based approach) are deployed to solve production planning problems for FMS in [5]. Production programming [6] is optimized using the economic model predictive control approach considering the energy market and its fluctuations. The study [7] proposes a methodology of estimation operational parameters using simulation for FMS. Also, a simulation-based optimization [8] method is proposed to achieve robust production planning and control for multi-stage systems with flexible final assembly lines. The study [9] solves the stochastic flexible job shop scheduling problem by an integrated genetic algorithm—Monte Carlo method. For the pre-mentioned studies, most of the papers related to FMS prove the concept through simulation.

In expanding various identification ability improvement systems for flexible production, the possibility of recognition systems using deep learning is expanding recently. In computer vision, deep learning is most actively applied and developed, and it is actually used in recognition systems for flexible production. Among them, object detection is to detect a specific element in an image. Object recognition models such as VGG-SSD [10], MobileNet-SSD [11], Faster-RCNN [12], and YOLO [13, 14] can be used in object detection. Object detection detects an object and its bounding box in an image. This deep learning model receives an image as input and outputs the bounding box, class list, and class confidence. Object detection is used in various industrial fields for autonomous driving, detecting abnormal situations in the medical field, and detecting defective parts in the manufacturing industry [15–18].

Internet of Things (IoT) [19–34] means that things around people are equipped with network functions and connected through the Internet anytime, anywhere. It is used to transmit data by connecting surrounding objects through the Internet, and to utilize the collected information and analyzed results. IoT is used in various service applications such as smart health [23–25], smart home [26–28], smart transportation [29–31], and smart manufacturing [32–34]. It was used in smart health applications for human activity recognition [23], energy expenditure estimation [24], and diabetic retinopathy detection [25]. In smart-home, IoT is applied to services such as indoor localization [26], device-free localization and activity recognition [27], and autonomous navigation [28]. As a smart transportation application, it was applied to short-term traffic prediction [29], traffic object tracking [30], and self-driving cars [31]. Finally, in the manufacturing field, computation burden calculation [32], surface inspection [32], and Wavelet-based fault diagnosis [34] were utilized through IoT. As mentioned in [22], the deep learning model using IoT derives powerful analysis results, however, requires high hardware requirements. In other words, there may be situations where it is necessary to use resource-limited embedded devices in terms of cost.

This paper intends to improve the quality of the painting process, which is the final finishing process, among the manufacturing processes for the production of high-pressure pipe joints that can connect pipes without welding, which is used for drilling. Existing 2-inches and 3-inches products are pre-produced and sold products, and workers are performing undercoating and 4–6 times of intermediate and top coating simultaneously. In this series of manufacturing processes, the painting and packaging process is currently being carried out manually. In particular, in the case of painting, the worker sprays the pipe, which is the work object, with a painting gun while considering the quality of the film thickness and is performed by simple repetitive work.

The IoT module developed in this paper is applied to FMS for the painting process of the high-pressure pipe to improve the poor working environment of the conventional manual process. The proposed painting process makes the coating quality uniform, dramatically improves productivity, and shortens the drying time. The proposed system recognizes nine types of pipes, three types of diameter (2 inches, 3 inches, and 4 inches) and three types of lengths (6ft., 8ft., and 10ft.), and informs the recognition results to the gripping robot and painting robot. Also, this paper presents the recognition algorithms for the classification of the pipe and a web-based monitoring system for monitoring results. By establishing a system that recognizes pipes regardless of the type of painted pipe, flexible production of the painting process is possible.

The advantage of the proposed method in this paper is to apply the appropriate technology [35] to the industrial site to avoid the problem of high hardware requirements of IoT modules using the aforementioned deep learning. Appropriate technology contains essential functions and has the potential to lower costs. In addition, it can continue to interoperate, extending existing and new equipment into modules. This ensures low operating costs as well as long-term use. Since the hardware of the deep learning model using machine vision camera and GPU is relatively expensive and difficult to apply to small and medium enterprises, it is necessary to develop a module with appropriate performance and reasonable price. Also, the proposed method is meaningful in applying the concept of FMS to the actual production line, unlike previous studies [1–9] that optimize FMS through simulation.

This paper is organized as follows. Section II describes the conventional painting process of the high-pressure pipe. Section III presents the proposed system and process. Section IV provides the method validation setup of the proposed methodology. Section V shows the results using the proposed process. In Section VI, the conclusion is presented.
2 Conventional painting process

A high-pressure (15,000 psi) pipe manufacturing for oil and gas drilling consists of material cutting, forging, heat treatment, machining, assembling, pressure testing, painting, and packaging, as shown in Fig. 1. The high-pressure (15,000 psi) pipe manufacturing for oil and gas drilling consists of material cutting, processing, assembling, pressure testing, painting, and packaging. This paper aims to improve the coating process during the manufacturing process of high-pressure pipes. As shown in Fig. 2, the painting process loads the pipe onto the conveyor belt, transports it, and then fixes the pipe. After manually spray painting the fixed pipe up to 4∼6 times, the pipe is transported on a conveyor belt for packaging. Additionally, Table 1 shows equipment corresponding to the pipe manufacturing process.

In this series of manufacturing processes, painting and packaging processes are currently being performed manually (the painting process in Fig. 1). In particular, in the case of painting, the operator sprays the paint onto the pipe, which is the work object, with the painting gun, while the operator repeatedly moves the nozzle gun to the left and right while watching the painting thickness.

The operator removes oil and foreign substances after processing and assembly before the middle/top coating work and performs the undercoat coating on the outer surface of the pipe (Pup Joint) using a brush. After that, the worker puts the pipe on the transport cart for intermediate/top coating, moves the product on the transport cart to the painting booth, rotates the pipe by hand about six times, and sprays the middle/top coating with a paint gun. After painting, the truck is moved to unload the product, and the operator completes the touch-up painting work with a brush on the unpainted section while visually checking it. After finishing the painting, the pipe is dried for more than 40 minutes to dry.

In this paper, we propose a flexible productivity improvement system for the painting process of high-pressure pipe using an IoT module. Through the proposed system, it is possible to improve the poor working environment of the manual painting process, make the coating quality uniform, dramatically improve productivity, reduce drying time, and reduce the amount of paint.

For automatically painting the pipe, it is necessary to recognize the length and diameter of the pipe. The existing method (Fig. 1) uses a proximity sensor to recognize the thickness and length as it passes by. The problem with the existing method is that it is easy to apply when the length...
and diameter of the pipe are limited. However, when the length and diameter are diversified, there is a problem that a proximity sensor must be installed, and each interface must be applied.

3 Proposed methodology

This study proposes a system and process that recognizes the thickness and length using a cost-competitive camera and IoT module to solve the previous section's problem. The process highlighted in yellow and orange in Fig. 2 is an improvement on the conventional process. In addition, the process was improved for flexible manufacturing of the pipe holding and manual painting parts in Fig. 1. To this end, the pipe classification process (yellow box in Fig. 2) and the classification process results are utilized for robot gripping and automatic painting of pipes (orange box in Fig. 2). Especially, this paper will focus on a system and process (yellow box in Fig. 2) for pipe classification using an IoT module and USB camera.

3.1 Proposed system hardware and its interfaces

As can be seen in Fig. 3, the system consists of an IoT module (Raspberry Pi module), an image acquisition unit (camera), and an edge TPU that infers object detection. In addition, the corresponding IoT module communicates with the PLC via Modbus TCP for controlling the robot and equipment so that the recognition result can be interfaced.

The IoT module includes a USB interface that can communicate with the Raspberry Pi camera, an HDMI port that can display images, and an Ethernet port that can communicate the pipe recognition result to other devices. Moreover, it includes a CPU and RAM for calculating the S.W algorithm and also includes wireless communication. The camera module transmits images to the IoT module.

| Table 1 | Conventional manufacturing process and equipment corresponding to the process |
|---------|--------------------------------------------------------------------------------|
| Process | Equipment                        |
| Cutting | CNC                               |
| Machining | CNC                              |
| Assembly | H/Machine                        |
| Pressure Test | H/T Machine                    |
| Painting | Manual                           |
| Packaging | Manual                          |
The specifications of the USB camera for acquiring the pipe image are specified in Table 2. The USB camera’s image sensor consists of a CMOS sensor, IMX291, and can acquire 2 Mega Pixels’ images. As for the frame rate, images of 30 fps can be obtained at an image size of 1920×1080, and this was continuously streamed. The focusing range of the camera is 1 meter to infinity. The camera for recognizing the pipe was installed at a position about 3 meters away from the position where the robot grips the pipe.

The IoT module is connected to the USB camera. The specifications of the IoT module are specified in Table 3. The OS of the IoT module is Raspbian Jessie, and the module’s operating program was developed based on python. The initial image obtained from the camera is transmitted in 1920×1080 resolution through proposed hardware interfaces to classify the pipe. Then, the IoT module and the camera are interfaced with USB, and the pipe classification is judged at least once every 5 seconds. The inference is enabled using OpenCV and Tensorflow lite. For Modbus communication, the “pymodbus” package is used.

Object detection is used in the proposed system to cross-check whether the pipe is positioned for robot gripping by the pipe image. However, object detection needs a lot of computation resources to infer results, so to speed up deep learning inference, the edge TPU module is interfaced with the IoT module. To operate the edge TPU module, at least 5V voltage and 500mA current must be supplied through USB. Power is supplied from the USB port of the IoT board. For inference, Edge TPU runtime is installed to enable Tensorflow Lite inference. In Tensorflow Lite, the inference speed of embedded systems can be increased by quantizing the weights of deep learning models. It is necessary to convert the 32-bit floating-point type of the existing weight or activation function to an 8-bit fixed-point type. It includes the process of re-learning the existing model, compiling it so that it can be used on the Edge TPU module, and deploying it on the module.

The result values obtained by the proposed system are linked to the robot and equipment PLC, and the result values can be checked on the web. As a Modbus TCP server in the equipment PLC, four addresses were used for Modbus TCP communication. The first address is a heartbeat address that flickers every second to check if communication is alive. The second address is an address that shows whether a signal from the proximity sensor gives signals whether the pipe has arrived at the gripper’s location. When the signal is “ON” at this address, the pipe classification algorithm is executed. The third address is the address that informs the pipe result recognized by the IoT module. Finally, the last address is an address indicating that classification is complete. When this address is changed to “ON,” the robot flexibly changes the gripper position according to the type of pipe and picks the pipe.

![Proposed system hardware and interfaces](image-url)
3.2 Proposed pipe classification process

As shown in Fig. 2, this paper proposes a process for classifying pipe types. First, the image is acquired through the USB camera, processed by the Raspberry Pi, and the pipe from the proximity sensor is reconfirmed through the object detection algorithm. Next, the proposed process detects a line in three regions of interest (ROI) and then determines the length and diameter of the pipe based on the result values of the line. Finally, the classification result is interfaced with the PLC of the robot gripping the pipe and the painting automation equipment to paint for the type of pipe flexibly.

3.2.1 Pipe classification algorithm

Figure 4 shows the pipe classification process. After the pipe is transferred from the conveyor belt, the proximity sensor is activated when the robot reaches the position to grip. When the proximity sensor is turned on, the camera image is captured. If it does not work, the conveyor continues to operate. After that, the object detection algorithm runs to cross-check whether the pipe has reached the gripping area.

When it is confirmed that the pipe is transferred from the conveyor in the proximity sensor and the object detection algorithm, the line detection algorithm (Fig. 5) proceeds in the ROI area of 3 parts. For example, a line is detected in Area 1, and it determines a 10 ft. pipe. If it is not detected, check the line detected in Area 2. If a straight line is detected in Area 2, it is an 8 ft. pipe. If it is not detected, a line is detected in Area 3. When a line is detected in Area 3, the pipe is classified as a 6 ft. pipe.

The diameter of the pipe is determined through the straight lines detected in Area 3. The diameter is calculated through the average of the y-coordinates of the detected straight lines. According to the diameter of each pipe, the threshold value of 4 inches is defined as $Th_{4inches}$, 3 inches is $Th_{3inches}$, and 2 inches is $Th_{2inches}$, respectively. For example, if the average of the y-coordinate is greater than $Th_{4inches}$, the 4 inches pipe is classified. Also, if the average of y-coordinates is smaller than $Th_{4inches}$ and larger than $Th_{3inches}$, it is classified as a 3-inch pipe. Moreover, if the image pixel of the y-coordinate average is smaller than $Th_{2inches}$, it is classified as a 2-inch pipe.

When the results of each pipe’s diameter (D) and length (L) come out, the classification results are sent to the PLC of the equipment for pipe gripping and automated painting in Fig. 2. Through this, the painting process can be applied flexibly even if the type of pipe changes on the conveyor.

3.2.2 Object detection algorithm

The model used in the object detection algorithm is the Yolov3-tiny model, and the $416 \times 416$ image is input. In this paper, the number of classes is one, and the name of the class is “PIPE.” The proposed system cross-checks whether the pipe has arrived at the grip position through the object detection model, thereby securing the reliability of the pipe recognition algorithm.

Yolo algorithm is one of the famous algorithms in one-stage object detection. Compared to two-stage object detection like Faster-RCNN [12], Yolo has a similar performance with fast inference speed. While 2-stage detection proceeds with bounding box regression and classification by making region proposals, the Yolo algorithm is fast because it does both at once using grid cells and anchors.

Yolo algorithm makes this possible by assigning an object to only one anchor box. Also, the Yolo algorithm finds the location of the bounding box with the location of the grid during prediction which is described in section 2.1 Bounding Box Prediction of [14]. The feature of the Yolo v3 model is that it is designed to produce 3 types of outputs in consideration of the scale of the object.
For supervised learning, the performance of the deep learning model is increased by reducing the loss. The loss function of Yolo-v3 is

\[
\text{Loss} = \lambda_{\text{coord}} \sum_i S^i \sum_j B \sum_{i,j}^{\text{obj}} \left[ \left( x_i - \hat{x}_i \right)^2 + \left( y_i - \hat{y}_i \right)^2 \right] + \lambda_{\text{coord}} \sum_i S^i \sum_j B \sum_{i,j}^{\text{obj}} \left[ \left( w_i - \hat{w}_i \right)^2 + \left( h_i - \hat{h}_i \right)^2 \right] + \sum_i S^i \sum_j B \sum_{i,j}^{\text{obj}} \left[ -o_i \log(\hat{o}_i) + (1 - o_i) \log(1 - \hat{o}_i) \right] \]

\[
+ \text{Mask}_{ij} \lambda_{\text{obj}} \sum_i S^i \sum_j B \sum_{i,j}^{\text{obj}} \left[ -c_i \log(\hat{c}_i) + (1 - c_i) \log(1 - \hat{c}_i) \right] + \sum_i S^i \sum_j B \sum_{i,j}^{\text{noobj}} \left[ -c_i \log(\hat{c}_i) + (1 - c_i) \log(1 - \hat{c}_i) \right],
\]

where \( S \) is the number of cells, \( B \) is the number of anchors, \( \lambda_{\text{coord}} \) is coefficient of the coordinate loss, \( \lambda_{\text{obj}} \) is coefficient of the objectness loss, \( \text{Mask}_{ij} \) is masking tensor, \( x_i \) is ground truth value of horizontal cell value \( \in [0, 1] \), \( \hat{x}_i \) is prediction value of horizontal cell value \( \in [0, 1] \), \( y_i \) is ground truth value of vertical cell value \( \in [0, 1] \), \( \hat{y}_i \) is prediction value of vertical cell value, \( w_i \) is ground truth value of the i-th object width, \( \hat{w}_i \) is prediction value of the i-th object width, \( h_i \) is the ground truth value of the i-th object height, \( o_i \) is i-th objectness of the ground truth, \( \hat{o}_i \) is i-th objectness of the prediction and \( c_i \) is the i-th class label of the ground truth, and \( \hat{c}_i \) is the i-th class label of the prediction, respectively. Also, \( \text{Mask}_{ij} \) is mapping of \( \{1, 0\} \) as a value of 1 if an object is present in grid cell \( i \) and the jth bounding box predictor is “responsible” for that prediction, and a value of 0 for other cases.

It has the meaning of proceeding with classification and location regression while checking where objectness and no-objectness exist in the above-mentioned three scales. This loss (1) is similar to the Yolo-v2 loss, however with differences that objectness, no-objectness, and classification loss all used binary cross-entropy, and the losses for \( x, y, w, \) and \( h \), the sum of square error was used.

### 3.2.3 Line detection algorithm

This subsection describes the line detection algorithm for classifying pipes (highlighted by the green box in Fig. 4). The algorithm for extracting lines from the ROI of the camera image consists of seven steps.

RGB to HSV conversion is performed to extract the corresponding color of the pipe from the RGB image. Since the pipe color is specified as red, only the color is extracted after HSV conversion and operation with the existing image. After the red color is extracted, it is converted into a gray...
image. Contrast-limited adaptive histogram equalization (CLAHE) [36] was applied in this study to increase the sharpness of gray images and to be robust against illumination under the pie surface condition. Next, apply Canny edge detection is applied to the image for detecting lines. After that, Gaussian blur is applied to remove the noise. Finally, a straight line is detected through the Hough transform. The detected straight line is used to determine the pipe class in three ROIs, as shown in Fig. 4.

3.2.4 PLC interface for flexible manufacturing system

The sequence program through image-based pipe recognition is interfaced with the painting equipment PLC, enabling the robot to install depending on the type of pipe. The development module and PLC interface are configured as shown in Fig. 3, and communication between the development module and the equipment PLC is configured through Modbus TCP communication. Four addresses are used for communication, the heartbeat indicating that communication is alive (ON/OFF every 1 second), the point at which the pipe comes from the conveyor belt to the place where the robot installs and starts to recognize the pipe type. When the pipe IN signal (transmitted from PLC to the development module) and pipe in signal comes in, the algorithm developed in the previous chapter determines the pipe type 3 times or more and consists of a pipe type and a signal indicating that recognition is complete. In addition, the result recognized by the development module is broadcast on the Web, and when a web page is accessed to a specific address, the recognition result and communication address value can be viewed.

4 Method validation setup

4.1 Image acquisition Hardware

As shown in Fig. 6, the proposed pipe classification system and process were installed and applied to the painting process. The yellow box shown in Fig. 6a is a system composed of an IoT module, an edge TPU, and a screen that can monitor it. The yellow box marked in Fig. 6b shows a USB camera to acquire an image and a jig to fix it. Finally, Fig. 6c is a router for Modbus TCP communication between the IoT module and PLC, and the result can be checked through the router through the web.

4.2 Object detection training results

A deep learning model for pipe recognition was established from the images acquired from the proposed IoT module and USB camera, and pipe location labeling was performed based on the acquired image. The Yolov3-tiny model was trained 2000 times, and the loss values were
The object detection result makes it possible to recheck whether the pipe is transferred from the conveyor belt to the position where the robot will grip.
5 Results and discussion

5.1 Deep learning-based object detection results

Figure 8 shows the result of object detection of the pipe. In Fig. 8a–c, the part marked with a purple box is the area recognized as a pipe. Pipe classification proceeds if the proximity sensor of the equipment that is held is recognized as ON when the area where the pipe is recognized is to hold the pipe to the robot gripper. In Fig. 8a–c, it can be seen that all pipes are detected.

5.2 Pipe classification results

Figure 9 shows the results of applying the process to a 3 inches, 10 ft. pipe, and 4 inches 10 ft. pipe. A blue box
indicates the ROI of the proposed system. The pipe type is determined after performing line detection in each area from the left to Area 1, Area 2, and Area 3. The first and second blue box from the left is a module that determines the length of the pipe. When a pipe exists, lines are detected. If the lines are detected in the first and second boxes, the proposed system results in a 10 ft. pipe. If the first straight line is not detected, and the second is detected, the system determines an 8 ft. pipe. Finally, if only the lines are detected in the 3rd box, it is recognized as 6 ft. pipe. The diameter is determined through the y-pixels of Area 3, and in the case of 2 inches, 3 inches, and 4 inches, the average position of the straight line of the corresponding pixel is the 6th, 7th, and 8th pixels, respectively.

In the case of 3 inches and 10 ft. (Fig. 9a) pipe, lines were detected in all areas and determined to be 10 ft. (3.048 m), and the average position of the straight line in area 3 was indicated as number 7 and determined as 3 inches (0.152 m). Also, in Fig. 9b, all straight lines were detected and classified as the longest pipe, 10 ft. (3.048 m), and the average position of the straight line was calculated as nine and classified as a 4 inch (0.203m) pipe.

5.3 Monitoring results

Figure 10 shows checking the pipe recognition process by accessing the result transmitted to the web from a mobile phone. On the left, after the pipe contacts the proximity sensor, before sending the pipe in signal from PLC, after the pipe in signal is turned on as shown in the middle of Fig. 9. After recognizing the pipe, as shown on the right side of Fig. 10, the third pipe (2 inches and 10ft) is recognized, and it is a picture that informs the PLC that the classification process is finished. Based on this web monitoring system, it is possible to monitor the results of pipe classification in real-time, and it can help workers make quick decisions in emergencies such as abnormal situations.
Performance results
The performance was measured by the unit production time of the proposed system. First, prepare six unpainted pipes. The time is measured from the moment the robot arm grabs the first pipe for painting. Second, after the robot is fixed to the rotating shaft for painting, painting begins. Third, the painted pipe arrives at the transfer conveyor. With the above procedure, the point at which all pipes arrive at the conveyor belt is the endpoint, and the total time is measured.

Table 4 shows the unit production time of the pipe painting process. The first row is the paint injection time, and the second row is the robot arm’s initial position movement time. It took 161.94 seconds to produce one pipe. Before applying the proposed module, the conventional productivity is manually produced 15 pipes per hour. The unit production time of the proposed system is calculated to be 22.23 pipes per hour. The productivity of the painting process has been improved by 48.2%.

Table 4 Process performance

| Pipe production number | 1    | 2    | 3    | 4    | 5 disc | 6    |
|-----------------------|------|------|------|------|--------|------|
| Paint injection time (s) | 139.34 | 139.93 | 139.09 | 139.21 | 139.02 | 139.59 |
| Robot arm’s initial position movement time (s) | 22.16 | 22.58 | 22.76 | 25.08 | 20.32 | 22.54 |

Conclusion
In this paper, an IoT module that recognizes nine types of pipes used in the painting process was developed. Recognizes nine types of pipes, three types of diameter (2 inches, 3 inches, and 4 inches) and three types of lengths (6ft, 8ft, and 10ft) used in the process, and sets the pipe type to the robot An IoT module that informs has been developed. The IoT module that recognizes the type interfaces with the painting robot, and the sequence control code for painting by type of pipe is executed on the painting robot, enabling one-coat painting to proceed. Through the automation of this module and the painting robot, nine types of pipes can be painted automatically. In other words, it is possible to paint products of different diameters and lengths through the robot work program flexibly.

The limitation of the proposed system is that it is difficult to apply in a system requiring high-speed inference. The proposed system is capable of inference at a rate of 3 fps. For higher inference performance, one can use a computer with a high computational power GPU or an edge device such as “Jetson Xavier”. Since the proposed modules’ price can be

![Fig. 10 Web monitoring results](image)
configured under 300$, it can be used in the manufacturing process of small and medium-sized enterprises with a cost burden. The advantage of the proposed module is that it has price competitiveness and can be used in processes using similar object size recognition.

**Funding** This study has been conducted with the support of the Ministry of SMEs and Startups as “Development of I-coating paint method and process innovation based on the shape recognition of ultra high pressure pipe”.

**Declarations**

**Consent to participate** Not applicable.

**Consent for publication** Not applicable.

**Conflicts of interest** The authors declare that they have no competing interests.

**References**

1. Fan Y, Yang J, Chen J, Pengcheng H, Wang X, Jianchun X, Zhou B (2021) A digital-twin visualized architecture for Flexible Manufacturing System. J Manuf Syst 60:176–201
2. Jenny L, Diaz C, Ocampo-Martinez C (2021) Non-centralised control strategies for energy-efficient and flexible manufacturing systems. J Manuf Syst 59:386–397
3. Hernandez JD, Cespedes ES, Gutierrez DA, Sanchez-Londoño D, Barbieri G, Abolghasem S, Romero D, Fumagalli L (2020) Human-Computer-Machine Interaction for the Supervision of Flexible Manufacturing Systems: A Case Study. IFAC-PapersOnLine 53(2):10550–10555
4. Long F, Zeiler P, Bertsche B (2017) Modelling the flexibility of production systems in Industry 4.0 for analysing their productivity and availability with high-level Petri nets. IFAC-PapersOnLine 50(1):5680–5687
5. Tiwchar MK, Saha J, Mukhopadhyay SK (2007) Heuristic solution approaches for combined-job sequencing and machine loading problem in flexible manufacturing systems. Int J Adv Manuf Technol 31:716–730
6. Jenny L, Dia C, Ocampo-Martinez C (2020) Optimal production planning for flexible manufacturing systems: an energy-based approach. IFAC-PapersOnLine 53(2):10461–10467
7. Florescu A, Sârbu FA (2017) Operational parameters estimation for a flexible manufacturing system. A case study. MATEC Web of Conferences 112(4):05008
8. Gylai D, Pfeiffer A, Monostori L (2017) Robust production planning and control for multistage systems with flexible final assembly lines. Int J Prod Res 55(13):3657–3673
9. Dosdoğru AT, GöçSken M, Geyik F (2015) Integration of genetic algorithm and Monte Carlo to analyze the effect of routing flexibility. Int J Adv Manuf Technol 81:1379–1389
10. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2015) SSD: Single Shot MultiBox Detector. arXiv:1512.02325v5
11. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Marco TW, Adam AH (2017) MobileNet: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861v1
12. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. arXiv:1506.01497
13. Redmon J, Divvala S, Girshick R, Farhadi A (2015) You Only Look Once: Unified, Real-Time Object Detection. arXiv:1506.02640
14. Redmon J, Farhadi A (2018) YOLOv3: An Incremental Improvement. arXiv:1804.02776
15. Zhao Z-Q, Zheng P, Shou-Tao X, Xindong W (2019) Object Detection with Deep Learning: A Review. IEEE Trans Neural Netw Learn Syst 30(11):3212–3232
16. Munir F, Azam S, Rafique MA, Sheri AM (2021) Moongu Jeon, Witold Pedrycz, Exploring Thermal Images for Object Detection in Underexposure Regions for Autonomous Driving. arXiv:2006.00821v2
17. Zhang H, Hong X (2019) Recent progresses on object detection: a brief review. Multimed Tools Appl 78:27809–27847
18. Kim JS, Choi KN, Kang SW (2021) Infrared Thermal Image-Based Sustainable Fault Detection for Electrical Facilities. Sustainability 13(2):557
19. SchulzRichard J, Popp S-H, Scharmer VM, Zaeh MF (2018) An IoT-based approach for energy flexible control of production systems. Procedia CIRP 69:650–655
20. Singh SK, Jeong Y-S, HuyPark J (2020) A deep learning-based IoT-oriented infrastructure for secure smart City. Sustain Cities Soc 60:650–655
21. Shah D, Wang J, He QP (2020) Feature engineering in big data analytics for IoT-enabled smart manufacturing – Comparison between deep learning and statistical learning. Comput Chem Eng 141:106970
22. Ma X, Yao T, Menglan H, Dong Y, Liu W, Wang F, Liu J (2019) A Survey on Deep Learning Empowered IoT Applications. IEEE Access 7:181721–181732
23. Hammerla NY, Halloran S, Ploetz T (2016) Deep, convolutional, and recurrent models for human activity recognition using wearables. arXiv:1604.08880
24. Zhu J, Pande A, Mohapatra P, Han JJ (2015) Using deep learning for energy expenditure estimation with wearable sensors. 2015 17th International Conference on E-health Networking, Application & Services (HealthCom)
25. Gulshan V, Peng L, Coram M, Stumpe MC, Wu D, Narayanaswamy A, Venugopalan S, Widner K, Madams T, Cuadros J, Kim R, Raman R, Nelson PC, Mega JL, Webster DR (2016) Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. Proc JAMA 316(22):2402–2410
26. Mohammad M, Al-Fujahia A, Guizani M, Oh J-S (2018) Semi-supervised deep reinforcement learning in support of IoT and smart city services. IEEE Internet Things J 5(2):624–635
27. Wang J, Zhang X, Gao Q, Yue H, Wang H (2017) Device-free wireless localization and activity recognition: A deep learning approach. IEEE Trans Veh Technol 66(7):6258–6267
28. Erol BA, Majumdar A, Lwowksi J, Benavidez P, Rad P, Jamshidi M (2018) Computational Intelligence for Pattern Recognition pp. 369-395
29. Zhao Z, Chen W, Wu X, Chen PCY, Liu J (2017) Recent progresses on object detection: a brief review. IET Intell Transp Syst 11(2):68–75
30. Li H, Li Y, Porikli F (2016) DeepTrack: Learning discriminative feature representations online for robust visual tracking. IEEE Trans Image Process 25(4):1834–1848
31. Bojarski M, Del Testa D, Dworakowski D, Firner B, Flepp B, Goyal P, Jackel LD, Monfort M, Müller U, Zhang J, Zhang X, Zhao J, Zieba K (2017) End to end learning for self-driving cars. arXiv:1604.07316
32. Li L, Ota K, Dong M (2018) Deep learning for smart industry: Efficient manufacture inspection system with fog computing. IEEE Trans Ind Inf 14(10):4665–4673
33. Park J-K, Kwon B-K, Park J-H, Kang D-J (2016) Machine learning-based imaging system for surface defect inspection. Int J Prec Eng Manuf-Green Technol 3:303–310
34. Wang J, Zhuang J, Duan L, Cheng W (2016) A multi-scale convolution neural network for featureless fault diagnosis. International Symposium on Flexible Automation (ISFA) p. 16544258
35. Jung WK, Kim DR, Lee HS, Lee TH, Yang J, Youn BD, Zontar D, Brockmann M, Brecherc C, Ahn SH (2021) Appropriate smart factory for SMEs: concept, application and perspective. Int J Precis Eng Manuf 22:201–215
36. Ju Sik Kim, Kim JS, Choi KN, Kang SW (2019) Improved method for enhancing dark images based on CLAHE and morphological reconstruction. Proc Comput Sci 165:391–398

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.