An Empirical Study on Process Management System using YOLO-Based Parts Recognition

Jongcheol Park¹, Wonho Jang² and Namhyun Yoo³

¹ The Company Co., Ltd., Room No. 303, 6-Dong, 21 Bongambuk 7-Gil, Masanhoewon-gu, Changwon-si, Gyeongsangnam-do 51347, Korea
² ForceTEC Co., Ltd., 6 Jungang-daero 228beon-gil, Uichang-gu, Changwon-si, Gyeongsangnam-do 51439, Korea
³ School of Computer Science & Engineering, 7 Kyungnamdaehak-ro, Masanhappo-gu, Changwon-si, Gyeongsangnam-do 51767, Korea
hyun43@kyungnam.ac.kr

Abstract. The process of assembling the engines used in medium and large ships appears to be a continuous production process, but it is not the process through which numerous workers perform work with automated equipment or simple assembly work, but rather a process through which skilled workers, who can effectively perform various tasks, perform multiple aspects of the assembly process. Due to this characteristic, it was difficult to assess the work process rate during the engine assembly process through real-time analysis of the process because it was difficult to shorten the lead time or improve the process, and most of stages of the process depend on manual reports or operators’ work data input. Therefore, in this study, we developed and applied a system capable of recognizing the current process rate by identifying specific engine parts using YOLO.

1. Introduction

In the process of assembling the engine used in medium and large ships, it is not possible for the operator to perform the work and simultaneously record the work contents in real time. In other words, writing the results for the work content cannot be done due to a time gap with the actual process progress status, and when the operator goes to eat or bathroom, it is recognized as 'not progressed' until related data is entered. In addition, in the event of an abnormality of assembly parts or various unexpected situations in the field, immediate response as well as preliminary measures are not possible except for the operator or colleagues around him, therefore, process cannot be managed in real life and this can cause decrease of productivity due to the increase of waiting time for production. In addition, the reliability of the data required to analyze the lead time of the assembly process is inevitably varied depending on the operator’s skill in creating a process status report because the operator directly writes the relevant information by hand and delivers it in writing. In addition, since there is a high concern about the safety of workers in the process of assembling a large ship, there is also a problem of dependency on inaccurate memory by making them record data after working hours rather than assigning other tasks other than assembly. In such an environment, if it is possible to eliminate unnecessary work, supply of parts for the next process and arrange the equipment in real time by checking the work progress of each process, the lead time will be shortened. Therefore, in this paper,
we intend to apply a production optimization system by developing a system that can predict the process progress rate through the recognition of the parts installed in the engine in real time between each process by installing a camera in the engine assembly workshop of STX Engine. In this paper, rather than covering the entire production optimization system, we plan to cover the monitoring of the process status using vision recognition technology.

2. Related work
There are algorithms such as CNN[1,2], YOLO[3,4], and SSD[5] as algorithms for applying image recognition techniques of parts for each location in the assembly process of medium and large ship engines. When using this system, as the operator can focus only on the assembly process and manage the progress rate for each work process in real time, it is expected to be able to perform process optimization work through analysis of lead time. In general, CNN can extract objects with high accuracy, but since it is impossible to process numerous images acquired by window detection sliding with CNN, R-CNN[6] algorithms have been developed for this, but there is a problem that it takes more time compared to other algorithms. Compared to CNN, YOLO divides each image into S x S grids and calculates the reliability of the grid, and the reliability reflects the accuracy when recognizing objects in the grid. The divided grid can obtain a boundary area having the highest recognition accuracy by adjusting the position of the boundary after calculating the reliability. Due to this method, it is recognized that it is twice as fast as that of other algorithms such as CNN. An algorithm that has a balance between CNN and YOLO is SSD. In this paper, since image recognition must be performed in the actual assembly process, we intend to perform image recognition using YOLO.

2.1. YOLO (You Only Look Once)
In this paper, YOLO is used to identify the parts of the ship engine to be assembled. As seen in [4], YOLO has the advantage of faster recognition speed compared to other algorithms. Existing Object Detection algorithms have two steps: 1) Region Proposal; and 2) Classification. But YOLO has a structure that removes Region Proposal and performs Object Detection at once. Figure 1 shows the structure of YOLO. After dividing the input image into the S x S grid area first, then B Bounding Boxes corresponding to the areas that may have problems in each grid area are predicted.

It appears in the form of (x,y,w,h), where (x,y) is the coordinates of the center point of the bounding box, and w,h are the width and height. The next step is to calculate Confidence, which means the reliability of the box. This is calculated after multiplying by IoU (Intersection over Union)
as shown in Equation 1, which means the ratio of Pr(Object) (probability of the existence of an object in the grid) and the overlapped area of the predicted box and the Ground Truth box.

\[ Box \text{ Confidence Score} = P_r(\text{object}) \times IoU_{\text{truth}} \quad (1) \]

Next, the probability that the corresponding class for C classes in each grid is calculated, and the basic formula is shown in Equation 2. The special point in this process is that when Object Detection is performed, the number of classes +1 (background) is always inserted for classification. Unlike other algorithms, YOLO uses a different classification method from other algorithms.

\[ Conditional \text{ Class Probability} = P_r(Class_i|object) \quad (2) \]

\[ Class \text{ Confidence Score} = P_r(Class_i) \times IoU = Box \text{ Confidence Score} \times Conditional \text{ Class Probability} \quad (3) \]

3. Implementation of process management system through YOLO-based part recognition

STX Engine, which is the organization where this research project is applied, is a company that produces mid to large-sized ship engines, and develops engines mounted on warships as well as engines for general merchant ships. Since there is a limitation that MB871 and MT881 assembly workplace can be recorded as original video but it cannot be leaked to the outside, this paper will show the results of application of related development contents using self-produced images and partially blurred images. Figure 2 shows the assembly site of the STX engine, where the engine is assembled in an area divided by a yellow box. In the engine assembly site, rails are basically installed for movement, and the engine is assembled on pallets installed on the rails. At the end of each assembly process, the pallet is pushed and moved in a different assembly process. The business sites to be applied in this study are MB871 and MT881 lines, and assembly is completed while moving from CAM1 to CAM6 on one rail.

![Figure 2. STX engine assembly site floor plan.](image)

Figure 3 shows the concept of managing the process through image recognition of each part in this paper. Cameras are installed above each assembly process, and the progress of the engine assembly process can be checked through recognition of parts in various locations of the engine. If parts A, B, and C are recognized in camera 1, it is judged that the assembly process is completed about 30%, and if only part I is seen in assembly position 3 of camera 2, it can be judged that the assembly process is completed about 70%. Installation of one camera was the first attempt, and it was confirmed that there was no difference in the overall recognition rate even if two cameras were installed.
Figure 3. YOLO-based parts recognition management process management concept diagram.

The language used to implement the process management system through YOLO-based part recognition is Python, and the application of YOLO was handled by using Google's Tensorflow[7].

- In order to maximize the performance of parts recognition, a lot of data was collected under various conditions. Image collection was performed by combining various conditions such as image size, shooting direction, and lighting, etc. and images were limited to those which can be identified with the naked eye.
- Using an image labeling program, humans did the labeling in a square shape based on the part classification code for each part.
- After labeling was completed, learning was conducted by using the image, about 8,000 times learnings were conducted, and the accuracy of the classified images was verified.
- When the object was not recognized or recognized incorrectly in the object recognition and processing results, only the corresponding image was additionally learned. At the same time, the recognition rate was improved by correcting the corresponding result by reviewing the error in the labeling image and the classification location data.

The completed YOLO model was applied to the actual process and the result was verified.

The process progress rate calculated based on the number of recognized parts for each process is as shown in Equation (4).

\[
R_{avr} = CR_{ppr} + n_{rp} \times \left( \frac{n_{rp}}{n_{tn}} \right)
\]  

\( R_{avr} \) is the process rate calculated based on the number of recognized parts in the current level, \( CR_{ppr} \) is the accumulated process rate until the previous assembly level, \( n_{rp} \) is the number of recognized parts, and \( n_{tn} \) is the number of parts to be recognized. Figure 4 and Figure 5 are showing the parts recognized in each step. Figure 6 is the implementation screen of process management system showing the current process rate calculated based on the number of recognized parts.

Figure 4. Assembly Location : MT881_M1.  
Figure 5. Assembly Location : MT881_ACC1.
4. Conclusion and future research
The ship assembly process of STX Engine, which produces medium to large-sized ships, consists of continuous production line, but most of the assembly processes are detailed processes in which the operator directly assembles parts rather than using dedicated assembly equipment. In addition, some parts are large and heavy, so safety accidents may occur during the assembly process. Moreover, when each step of the assembly process is completed, it is possible to input the work content depending on the worker's memory and the situations that occurred in the assembly process, therefore, it is difficult to calculate the proper lead time or perform process improvement through the analysis of the work content in the process. Therefore, in this paper, it is possible to estimate the process rate for each workshop in real time through recognition of parts that can grasp the progress of each step after installing a camera on each assembly workshop. In the future, it is planned to increase the recognition rate from the current 90% level to 99% through acquisition and learning of work site images, improvement of labeling, optimal location for image acquisition, and YOLO algorithm correction in various environments.

5. References
[1] Albawi S, Mohammed T and Al-Zawi 2017 Proc. Int. Conf. on Engineering and Technology (Antalya) (IEEE) p 1.
[2] Abiodun O, Jantan A, Omolara A, Dada K, Mohamed N and Arshad H 2018 J. Heliyon 4 e00938.
[3] Redmon J., Divvala S., Girshick R. and Farhadi A 2016 Proc. Int. Conf. on Computer Vision and Pattern Recognition (Las Vegas) vol 1 (IEEE) pp 779-88.
[4] Shafiee M, Chywl B, Li F and Wong A 2017 Fast YOLO: A Fast You Only Look Once System for Real-Time Embedded Object Detection in Video arXiv (2017)1709.05943.
[5] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C and Berg A 2016 Lecture Notes in Computer Science 9905 21.
[6] Girshick R 2015 Proc. Int. Conf. on Computer Vision (Santiago) vol 1 (IEEE) pp 1440-1448.
[7] Dillon J., Langmore I., Tran D., Brevedo E., Vasudevan S., Moore D., Patton B., Alemi A., Hoffman M. and Saurous R J. 2017 TensorFlow Distributions arXiv (2017)1711.10604.

Acknowledgments
We thank the Software Convergence Cluster 2.0 funded by Ministry of Science and ICT, Gyeongsangnam-do, and National IT Industry Promotion Agency.