IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 9, 2019, accepted December 22, 2019, date of publication December 25, 2019, date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962131

Robust Design of Detecting Contaminants in Façade Cleaning Applications

JISEOK LEE[1], GARAM PARK[1], YECHEOL MOON[1], SUNGON LEE[2], (Member IEEE), AND TAEWON SEO[1], (Member, IEEE)

[1]School of Mechanical Engineering, Hanyang University, Seoul 04763, South Korea
[2]School of Electrical Engineering, Hanyang University, Ansan 15588, South Korea

Corresponding authors: Sungon Lee (sungon@hanyang.ac.kr) and Taewon Seo (taewonseo@hanyang.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Science and ICT for the First-Mover Program for Accelerating Disruptive Technology Development under Grant NRF-2018M3C1B9088328 (2018M3C1B9088331 and 2018M3C1B9088332) and in part by the Hanyang University under Grant HY-2019.

ABSTRACT As the number of high-rise buildings is increasing, more methods of exterior-wall cleaning are being developed. There are a few models based on artificial intelligence that determine the type and level of contamination primarily by moving the cleaning area. In this study, we propose an system using YOLOv3 algorithm, color-detection, to install on façade cleaning robot and brightness-discrimination. There are three types of contaminant-detection parameters: size, color, and brightness, and these parameters are subjected to a robust optimization process to maintain a constant detection rate under different conditions. The three parameters are determined via Taguchi method with signal to noise ratio and noise factors. An environment for algorithm testing is established, and artificial contamination is implemented on the specimen. A field test with the detection algorithm shall be performed in the near future.

INDEX TERMS Façade cleaning, image processing, service robot, Taguchi method optimization, parameter optimization, detection algorithm.

I. INTRODUCTION

In recent years, as the number of high-rise buildings has increased, the market demand for exterior-wall cleaning service for such buildings has also increased. Reference [1] Because the task of cleaning such walls is very dangerous for the workers, it is necessary to replace them with robots. To this end, various exterior-wall-cleaning robots have been developed [2]. By attaching these robots to a system that can detect contaminants on an exterior wall, we can perform numerous tasks. Recently, the gondola-embedded façade cleaning robots are proposed [3]–[6]; however, the detection method of the contamination is not researched yet. Based on the type of contaminant, an appropriate cleaning solution can be automatically identified, and the contamination of the exterior wall can be transmitted to a central management system that shall automatically monitor the cleanliness of the wall.

To develop this exterior-wall contaminant-detection system, we searched for a surface-contaminant-detection method. This detection method had to be compact because it were to be installed on a mobile platform attached to the exterior walls of buildings. In addition, it must be possible to detect all types of contaminants including dust, rusty stains, and avian feces. There are two methods of detecting contamination of exterior-wall surface. First method is irradiating an IR ray (infra red ray) and analyzing the reflected light [7]. However, liquid contaminants are not the only type of contaminants on the exterior walls of buildings. This can give rise to an error in the detection result, thereby making this method difficult to apply. There is a surface-contaminant-detection method that uses Mini Raman Lidar [8]. However, the system is too large to be installed on a mobile platform attached to the exterior walls of buildings.

Because workers in the field identify contaminants with the naked eye, we propose a machine-vision system that uses CNN(convolutional neural network) and image processing [9]. In this study, the contaminants on the exterior walls of buildings are classified into three types: object, area, and particle. The YOLOv3 module detects object-type contaminants; irregular contaminants such as avian feces are detected using this module [10]. Next, area-type contaminants are detected using the color-detection module; contaminants that exist in areas of the same color such as rusty stains are detected

The associate editor coordinating the review of this manuscript and approving it for publication was Donatella Darsena.

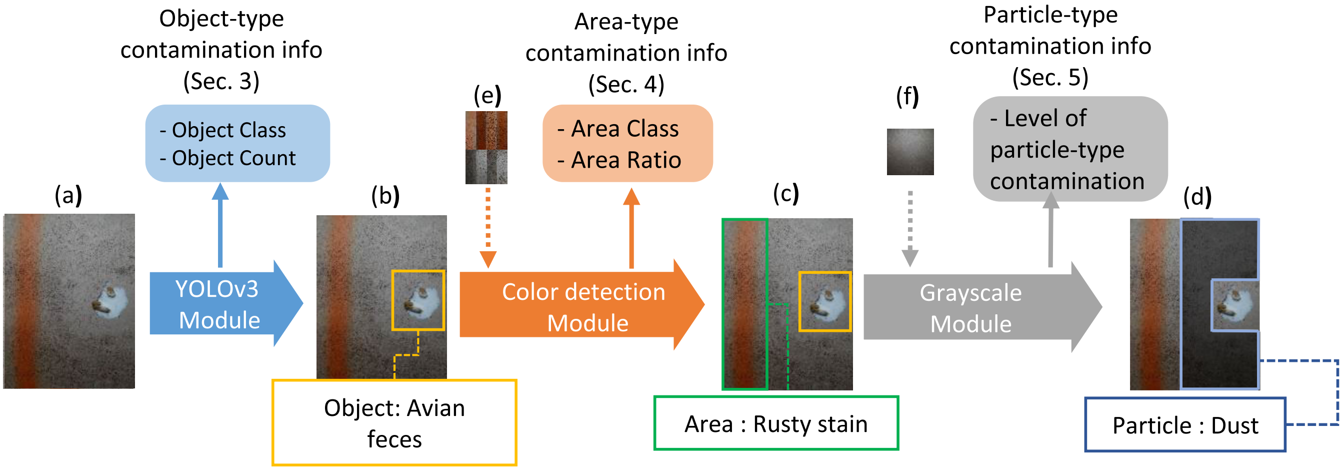


FIGURE 1. **Contamination detection system overview: (a) original image, (b) image pass after YOLOv3 module result, (c) image pass after color-detection module, (d) image with marked area processed in grayscale module. (e) background and color area sample provided in advance, and (f) reference image of clean facade provided in advance.**

using the HSV color space and flood fill algorithm [11]. Finally, particle-type contaminants are detected using the grayscale module. The darkness level of a picture is computed by comparing its average brightness in a certain area with the reference average brightness of the clean exterior wall. Dust build up on the surface changes the average brightness of the picture. using this point, grayscale module indirectly measures the amount of dust on the surface. A previous study can be referred to for more details [9].

In this study, we use the Taguchi method to tune the system parameters such that the machine-vision system described earlier is robust to various external brightness levels and distance from the facade [12], [13]. Parameter tuning for various engineering problems was performed using the Taguchi method [14]. The removal rate of EN24 steel used as a workpiece material was optimized. In addition, the Taguchi method can be used to optimize end-milling parameters [15].

Inside the system, three modules process information in a sequence, and the system parameters of each module are different. Because the processing of information is sequential, the first step involves the tuning of parameters of the You Only Look Once(YOLO) v3 module. Secondly, with the system parameters of the YOLOv3 module fixed, we proceed with the tuning of system parameters of the next module (i.e., the color-detection module). Finally, we tune the system parameters of the third module (i.e., the grayscale module) with the system parameters of the previous two modules fixed.

This paper is organized as follows. First, we define the problem in Section 2. We briefly introduce the machine-vision system that we wish to optimize and describe the Taguchi method, test bench, and specimen. Sections 3 through 5 define the parameters to be optimized and the objective function of each module. The design of each experiment using the Taguchi method and the selection of orthogonal arrays are discussed. Section 6 conducts experiments using orthogonal arrays and presents the tuned parameters. Section 7 tests the system using the tuned system parameters and verifies the results of the optimization. Section 8 presents the conclusion.

II. PROBLEM DEFINITION

In this study, Taguchi method is used to tune the parameters of the contaminant-detection machine-vision to deliver a constant level of accuracy under various conditions. It must be possible for this machine-vision system to be installed on a mobile platform that moves on the facade of a building. Therefore, we determine a system parameter that effectively performs contaminant detection when an image is taken, irrespective of the brightness of the image or the distance between the facade and the camera.

A. DETECTION SYSTEM OVERVIEW

The structure of the machine-vision system for parameter tuning in this study is shown in Figure 1. The system consists of the YOLOv3, color-detection, and grayscale modules. First, the YOLOv3 module detects object-type contaminants and make a box around of each detected objects, the bounding-box. The detected object bounding-box information is sent to the next module, and the size information of the box is sent to next module as object-type contaminant information. The color-detection module extracts information of the region that has a specific color by using the HSV color space for the remaining regions out of the bounding boxes.

The bounding-box information of the detected area is sent to the next module, and the size information of the detected area is sent as area-type contaminant information in proportion to the overall size of the image. In this process, we use a sample image for the background and color area provided in advance. The grayscale module calculates the average value of brightness by converting the remaining areas except for the bounding box detected by the previous two modules into grayscale images. Thereafter, the amount of particle-type contaminant is estimated based on the darkness of the picture compared with the reference value previously stored. For more details, please refer to a previously published article [9].

B. USER CONDITION DEFINITION

The optimization system must be installed on a mobile platform attached to the building facade. The setting of the camera is fixed, but the height and brightness may be different. Therefore, it is necessary to build a system that can perform robust detection under various height and brightness conditions. Therefore, we have selected these as user conditions.

C. ROBUST OPTIMAL PARAMETER DESIGN

Robust optimal parameter design is an experiment aimed at determining the value of the design variable that maintains the best condition without affecting the accuracy of the system even if the user conditions change. We choose the Taguchi method to perform the optimization. As detailed in this book [12], the robustness of the system in the case of this method is expressed as the signal-to-noise ratio (*SNR*). Tuning is performed by selecting the design variable with the highest *SNR*. To define the *SNR*, we first need to create an objective function that can effectively evaluate the results of the system. For a larger-the-better problem, *SNR* can be defined as

$$SNR = -10 \log \left| \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{y_i} \right)^2 \right| \tag{1}$$

where n is the total number of times the experiment is conducted, and y_i is the value of the objective function of the i-th experiment.

For a smaller-the-better problem, *SNR*can be defined as

$$SNR = -10 \log \left| \frac{1}{n} \sum_{i=1}^{n} y_i^2 \right| \tag{2}$$

Because conducting experiments for each design variable shall consume a considerable amount of time, an orthogonal array is used. We do not need to use every combination of design variables. Instead, the experiment can be conducted with an orthogonal array determined by the level and number of design variables. Next, we collect the results of the experiments, calculate the *SNR*, perform sensitivity analysis, and tune the design variables through additional experiments. The orthogonal array is given by

$$L_E \left(n^c \right), \tag{3}$$

where n is the level of each variable, c is the number of variables, and E is the number of rows in the orthogonal array.

$L_4(2^3)$, $L_9(3^4)$, and $L_{27}(3^{13})$ are commonly used orthogonal arrays. An orthogonal array depends on the number of chosen design variables and their level, and the number of correlation variable combinations used when a strong correlation between two specific variables is suspected. The degree of freedom of an orthogonal array must be greater than the degree of freedom of the experiment. For an orthogonal array $L_E(n^c)$, the *DOF* (degree of freedom) can be defined by

$$DOF = E - 1 \tag{4}$$

The experiment consists of a design variable with two levels and b design variables with three levels $(2^a \times 3^b)$. The *DOF* of the correlation variable combination with x-level variables and y-level variables is

$$Corr_{xy} = (x-1) \times (y-1) \tag{5}$$

$$DOF = (2-1)\,a + (3-1)\,b + \sum Corr_{xy} \tag{6}$$

To determine the orthogonal array of an experiment that contains a combination of correlation variables, we must refer to a linear graph that demonstrates the relationships among the columns of the array. After identifying the location where the correlation variable combination can be placed in the linear graph, we select the orthogonal array. Linear graphs of various orthogonal arrays are detailed in a book [13].

After conducting the designed experiment, we calculate the average *SNR* for each level of each design variable, and then express it as a graph for sensitivity analysis. This allows us to analyze the significance of each design variable and the way each affects the outcome.

D. TEST BENCH AND SPECIMEN

In order to carry out the robust design as described above, specimens and test benches were fabricated to modify each control factor in the learned network. The test bench was made of aluminum profiles in order to provide a uniform image of the specimen. The structure of the test bench shown in Figure 2 is composed of a holder for fixing the camera, a light source, bracket for adjusting the height, and table for photographing the specimen. The specimens were placed on a photography table, and six pictures were taken according to the user conditions of two heights and three brightness levels. Additional pictures were taken at 90°, 180°, and 270° to ensure reliability.

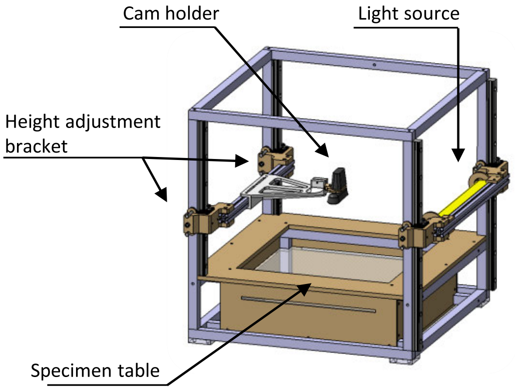


FIGURE 2. Test bench for specimen capture.

As shown in Figure 3, the specimens were {Bird feces, Rust, Dust}, which were grouped into three groups:

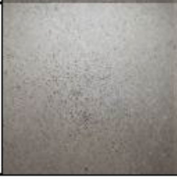
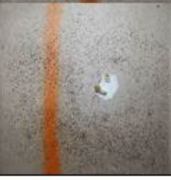
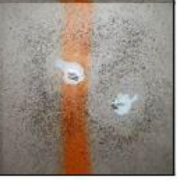
	Avian feces 0 Rust 0 Dust 1	Avian feces 1 Rust 1 Dust 2	Avian feces 2 Rust 1 Dust 3
Light gray plate			
Gray plate			

FIGURE 3. Specimens made for experiment.

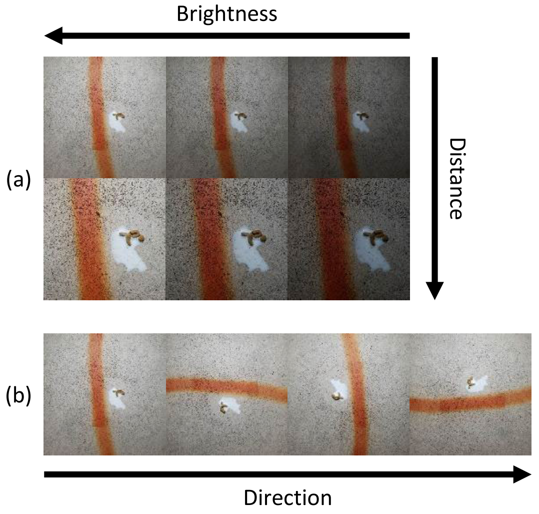


FIGURE 4. Specimen images used in experiment: (a) three levels of brightness and two levels of distance for one specimen, and (b) four directions taken for one specimen.

{0,0,1Cup}, {1,1,2Cup}, and {2,1,3Cup}. These contaminated specimens were fabricated into two color sheets, and six specimens were produced. The test specimens were tested for brightness (three steps) and distance (two steps) between the specimen and the camera via the user condition factor. To ensure the diversity and reliability of the study, each specimens of four degrees of 0, 90, 180 and 270 degrees were made. Thus, 24 pieces of picture data were taken from one piece of paper, and a total of 144 pieces of picture data were obtained. Figure 4 shows pictures taken for each specimen. We performed a robust optimal design procedure by changing each system parameter of each module. Then, this process was repeated through an *SNR* analysis to derive the optimal result.

III. YOLOv3 MODULE EXPERIMENT DESIGN

This module detects object-type contaminants such as avian feces that can be attached to a wall by embedding the YOLOv3 algorithm [10]. As you can see in Figure 5, this module detects the object-type contaminant in three steps and sends the class and count of the object information to the next module. In addition, the bounding-box information of the detected object is sent to the next module (the color-detection module) and is used to prevent the next module from rescanning the previously examined area. The YOLOv3 module first passes an image through an internal trained deep neural network (DNN) to print the object's class and bounding boxes. Next, we combine the overlapping boxes for an object into one by using an algorithm called non-maximum suppression (NMS). Finally, if the confidence level assigned to each bounding box is less than a certain threshold, then the box is deleted to derive the final detection result.

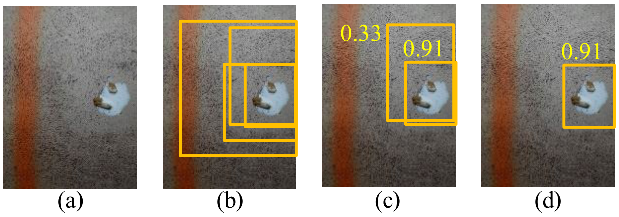


FIGURE 5. Results of each YOLOv3 module process step: (a) original image, (b) bounding boxes after passing DNN inside, (c) image after NMS processing, and (d) final image after confidence-level filtering.

A. DESIGN VARIABLES

1) NMS THRESHOLD (A)

After passing through the CNN inside YOLOv3, there are a number of bounding boxes. At this time, NMS is used for merging overlapping bounding boxes for an object into one. The NMS process first sorts the bounding boxes of a particular class in order of confidence. Then, starting with the bounding box with the highest confidence, the bounding box with Intersection over Union (IoU) above a certain NMS threshold is deleted given that it detects same object [16]. This process is repeated until the end, and then the box with the highest confidence is used among the remaining bounding boxes except the box that was the reference, and the process is repeated. At this time, NMS threshold is a real number between 0 and 1. Too large a value can result in multiple bounding boxes for one object, and a small value can result in only one bounding box for multiple objects. Therefore, this value is set as a system parameter called design variable A.

2) CONFIDENCE THRESHOLD (B)

After the NMS process, various bounding boxes remain. Among them, there may be bounding boxes that do not match the actual results. In this case, the confidence level indicates the probability of accurate object in the bounding box. So better accuracy can be obtained by deleting the bounding boxes that have a value below a certain confidence threshold. The confidence threshold is a real number between 0 and 1. If this value is too large, then the correct bounding box may be deleted, and if it is too small, then the incorrect bounding box may remain. Therefore, this value is set as a system parameter called design variable B.

B. ORTHOGONAL ARRAY

We set the NMS threshold to a three-level design variable A and the confidence threshold to a three-level design variable B. After the NMS process, the confidence threshold was used for the result, so we planned an experiment to consider the correlation variable combination of A and B. The *DOF* of this experiment is $(3 - 1) \times 2 + (3 - 1) \times (3 - 1) = 8$. Therefore, the*DOF* of the orthogonal array in this experiment must be at least 8. Because there is one correlation variable combination, we need to ensure that this combination can fit in the linear graph. Therefore, we selected orthogonal array $L_9(3^4)$ in this experiment. It has a *DOF* of 8, and it is confirmed that a pair of correlation variable combinations can be put through a linear graph. A linear graph is available in [13]. More detailed information about orthogonal arrays and linear graphs appears in Appendix 1.

C. DESIGN OBJECTIVE FUNCTION

The YOLOv3 module that we wish to optimize has two-part outputs. One is the class and count of objects that shall be passed externally as object-type contaminant information. The other is a bounding box that shall be passed to the next module. Therefore, evaluating the detection result of the YOLOv3 module requires a method of merging the accuracy of the class, count, and bounding box into one result.

The information about each object class was examined, and then the average was calculated. In this case, we did not go through this process because there is only one class: avian feces. IoU was selected as an indicator for the bounding box. IoU is a measure of the similarity between two bounding boxes. It uses the ratio of the union of the true and the detected bounding boxes and the intersection areas of the two boxes. Therefore, when the intersection area with the true bounding box is too small or the detected bounding box is too large, the IoU value is reduced. In this paper, IoU values between all true bounding boxes and detected bounding boxes are used.

We want to merge accuracy of area with information about the number of bounding boxes. The more the number of true bounding boxes same with detection boxes, the better the result. In addition, each bounding box must be in a meaningful position. To measure this value, we introduce three numbers: TB_c (true box count), DB_c (detected box count), CB_c (correct box count). TB_c is the number of true bounding boxes, and DB_c is the number of detected bounding boxes. CB_c is the number of boxes pared to TB among the detection bounding boxes. The method of calculation that was applied was a modification of the evaluation method of PASCAL VOC, the object detection algorithm competition [13]. In PASCAL VOC, if more than one bounding box exists for the same object, all the boxes are invalidated, but we decided to leave only one box.

For each detected bounding box, if there is no intersection area with the true bounding box, or if there are two or more, with one correct bounding box, that detected bounding box cannot be true bounding box. In addition, if the IoU with the true bounding box and detected bounding box is not more than 50%, then it is not the correct bounding box. Lastly, two or more detected bounding boxes for a true bounding box have an IoU value of 50% or more. If they overlap, only one is considered a correct bounding box. An example of a correct bounding box is shown in Figure 6; in (a), the IoU value for a true bounding box with an intersection area did not exceed 50%, whereas (b) had intersections with two or more true bounding boxes at the same time, and (c) had no intersection. Because (d) and (e) have an IoU value of 50% or more for one true bounding box, only one of them is treated as a correct bounding box.

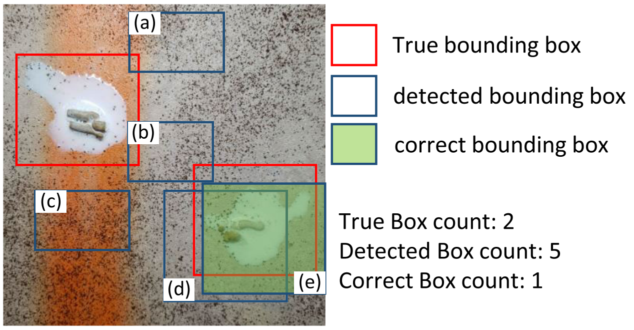


FIGURE 6. Example of TB_c, DB_c, and CB_c. True box count is number of true bounding boxes. Detected count is number of detected bounding boxes. Correct box count is number of correct boxes matched to true bounding box.

Next, a new index, the count accuracy ratio $CAR\,(\alpha, \beta)$, was defined to merge IoU and count information into one index called $errIoU_c$(error of IoU with count). At this time, $CAR(\alpha, \beta)$ is as follows:

Only if $(\alpha \geq \beta)$,

$$CAR\,(\alpha \geq \beta) = \left(\frac{\beta}{\alpha + \varepsilon}\right)^{(\alpha-\beta)/\alpha}, \quad \varepsilon = 2 \tag{7}$$

where $(\alpha - \beta)/\alpha$ is the error ratio, and ε is a spacer constant.

As can be seen in Figure 7, the *CAR* index always has a value between 0 and 1 depending on the error ratio. Even if they have the same error ratio, the larger the value of α, the larger the result value of the *CAR* index. In object detection, detecting half of two objects in one screen and half of 10 objects in one screen cannot be treated in the same way. Because the latter is a more difficult task, the greater the total number of objects, the better the output, even under the same error ratio. Depending on the spacer constant, the graph lines for α values are bent downward and become spaced apart. We determined the *CAR* value to be 0.5 when one of the two objects is detected (i.e., *CAR* (2,1) is 0.5, and the spacer constant is 2).

Using the *CAR* index, IoU, $\boldsymbol{TB_c}$, $\boldsymbol{DB_c}$, and $\boldsymbol{CB_c}$ are integrated to define an object function called $\boldsymbol{errIoU_c}$.

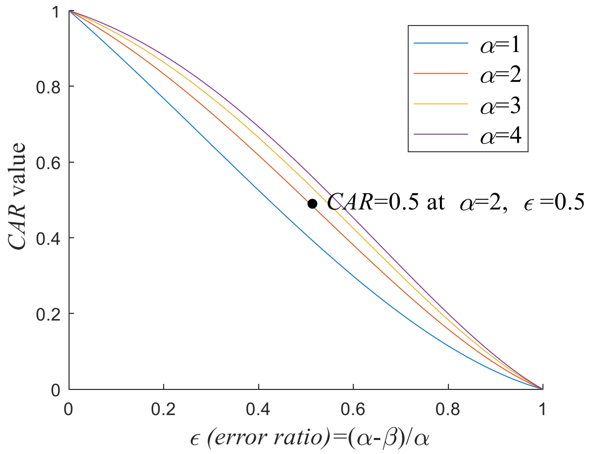


FIGURE 7. $CAR(\alpha, \beta)$ plot with $\alpha = 1$–4. $CAR(2, 1) = 0.5$.

The calculation method is as follows:

$$\begin{aligned}
&\text{If } (DB_c \neq 0 \text{ and } TB_c \neq 0)\\
&\quad errIoU_c = 1 - \text{IoU} \times CAR(TB_c, CB_c) \times CAR(DB_c, CB_c)\\
&\text{Else if}(DB_c = TB_c = 0)\\
&\qquad errIoU_c = 0\\
&\text{Else if } (DB_c = 0 \text{ and } TB_c \geq 1)\\
&\qquad errIoU_c = 1\\
&\text{Else if } (DB_c \geq 1 \text{ and } TB_c = 0)\\
&\qquad errIoU_c = 1 \qquad (8)
\end{aligned}$$

Because this is a smaller-the-better problem in the Taguchi method, it is calculated using Equation (2) after the experiment.

IV. COLOR-DETECTION MODULE EXPERIMENT DESIGN

This module converts an image into the HSV color space [11] and extracts a part with a specific color. This module is mainly used to detect area-type contaminant (ex-rust stain). You can see the results of each process in Figure 8. First, the image is converted to the HSV color space to find the hue and saturation values of each pixel. Next, we map the hue value to the hue detection range we want for each pixel with a value above the specified saturation for each pixel. The hue detection range is automatically determined using sample pictures of the area with the color to be detected for a specific brightness and sample pictures of the rest of the background.

At this time, hue margin value is given for hue detection range to cope with noise generated when shooting with a real camera. Then, the image is down-sampled to a certain size and the result is stored in a flag map. The sampling of one point is accomplished by using a median blur algorithm for the mapped hue value [18]. This blurs various color-difference noises caused by the camera characteristics. Finally, the flood fill algorithm is applied to the flag map to create the area-bounding-box information for the detected color area. This only checks the area other than the object bounding box detected by the YOLOv3 module. Last, the sum of the number of detected pixels in the area bounding boxes in proportion to the number of pixels of the entire picture is calculated and transmitted to the outside as area-type contamination information. Please refer to [9] for more details.

A. DESIGN VARIABLES

1) HUE MARGIN (C)

After extracting the hue value of the photo using HSV color space, it is used to create a hue depth map. In this module, we receive a sample image of the area-type contaminant and its background image at initialization. Then, we calculate and save the hue range we want to find. The determined hue-range information at initialization is used to create the hue depth map. By slightly increasing the range of both ends of the hue range, we tried to solve the color difference noise that occurs in the real-time detection image that was not in the sample image. If the range is increased too much, it will recognize too many colors; and if it is too small, it will be vulnerable to color-difference noise. Therefore, we set this range-increase value as a system parameter called design variable C.

2) MEDIAN FILTER SIZE (D)

The hue depth map still has noise from the camera. As you can see in Figure 8(c), there is a significant amount of fine noise left when shooting an image. If you pass through the color-detection process using this image, then the result will be very unstable. We use median filters to drastically reduce these noises and stabilize the results [18]. Adjusting the length of one side of the median filter will change the performance. If this value is too large, then the processing time will be long, and the boundary will become blurred. If this value is too small, then the noise of the hue depth map will remain. Therefore, we set this value as a system parameter called design variable D.

3) DOWN-SAMPLING SIZE (E)

Earlier, we used a median filter on the hue depth map to reduce noise. The next step, the process of creating a bounding box using the flood fill algorithm, is greatly affected by the size of the image. Area-type contaminants are often unclear in boundary areas, unlike in problems such as object segmenting. Therefore, there is a relatively low need for precise measurement of the boundary of the area. Thus, down-sampling reduces the size of the flag map from the hue depth map and reduces the number of times the median filter is used. Therefore, we specified the number of pixels between pixels that are down-sampled. If this value is too large, then the area information of the area-type contaminant begins to be crushed in the detection result. If the value is too small, then the process time is increased. Therefore, we set this value as a system parameter called design variable E.

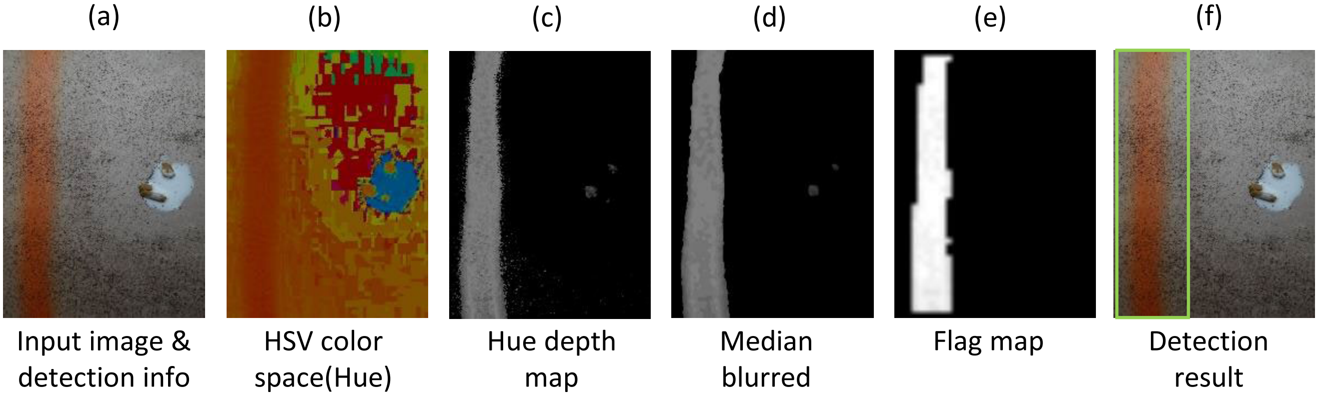


FIGURE 8. Results of each color-detection module process step: (a) bounding box after YOLOv3 module process, (b) detected color using HSV color space, (c) hue depth map, (d) median blurred hue depth map, (e) flag map, and (f) final detection.

B. ORTHOGONAL ARRAY

We set the hue margin to a three-level design variable C, the median filter size to three-level design variable D, and the down-sampling size to a design variable E. Since the hue depth map was created and the median filter was used simultaneously during the down-sampling process, we planned an experiment that considered the correlation variable combinations of C and D, D and E, and C and E. The *DOF* of this experiment is (3 - 1) × 3 + (3 - 1) × (3 - 1) × 3 = 20. Therefore, the orthogonal array in this experiment should have at least 20 *DOFs*. It should have a linear graph that contains three correlation variable combinations. We selected $L_{27}(3^{13})$ as the orthogonal array for this experiment. It has a *DOF* of 26, and it is confirmed that three pairs of correlation variable combinations can be put through a linear graph. More detailed information about orthogonal arrays and linear graphs appears in Appendix 2.

C. DESIGN OBJECTIVE FUNCTION

There are two parts of information from the color-detection module that we want to optimize. First, the area class and area ratio are sent to the outside as color-area detection info. Second, we create a bounding box for the color area and pass it to the next module so that the grayscale module cannot calculate the area of the other detection module. Therefore, this object function must combine three pieces of information into one.

The area class was not calculated separately because only one class (rusty stain) exists on the specimen. The area ratio is the ratio of the area occupied by the area-type contaminant to the total area. In this case, the ratio of the area of the area-type contaminant to the total area of the photograph previously calculated for the specimen is called the true area ratio TA_r. The area ratio of the specimen calculated using the system parameter determined by using the orthogonal array is called the detected area ratio DA_r, and each is calculated as follows:

$$TA_r = \frac{D_p}{T_p} \tag{9}$$

where D_p is the detected pixel count, and T_p is the total pixel count of image.

$$DA_r = \frac{D_{fp} \times M^2}{TA_r} \tag{10}$$

where D_{fp} is the detected flag map pixel count, and M is the median filter size.

In this paper, we defined an index called the Area Accuracy Ratio *(AAR)* to merge TA_r and DA_r into one value. Figure 9 shows the plot of the *AAR* index. The calculation method is as follows:

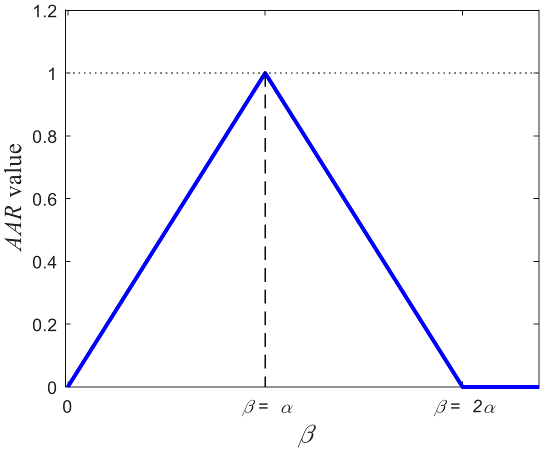


FIGURE 9. Graph of $AAR(\alpha, \beta)$ by β.

If $(0 < \beta \le 2\alpha)$

$$AAR\,(\alpha, \beta) = 1 - \left|\frac{\alpha - \beta}{\alpha}\right| \tag{11}$$

Else

$$AAR\,(\alpha, \beta) = 0 \tag{12}$$

If the desired reference value is put into α, and the detection result is put into β, then the closer the values of α and β, the closer they are to 1; and the farther apart they are, the closer they are to 0. By using this, two values can be merged into one $AAR(TA_r, DA_r)$.

IoU is used as an index for the accuracy of the bounding box for area-type contaminants. It does not need to compare box counts as we did with the YOLOv3 module. Thus, we did not use any additional indexes. The IoU can be calculated for the true bounding box and the detected bounding box as a value between 0 and 1.

Among the system parameters to use the Taguchi method in the color-detection module, the median filter size and subsampling size have a tradeoff between the detection accuracy and processing time. Thus, if the processing time is not considered when defining an object function, then the optimality and results of both parameters will go to the extreme. Therefore, information about the processing time should be merged with the object function. The detection system introduced in this paper processes the color-detection module after passing through the YOLOv3 module when a new image arrives. Owing to the nature of the module, the processing time of the YOLOv3 module is very long at around 3200 ms. Therefore, the processing time of the color-detection module is not to exceed 32 ms, which is 1% of 3200 ms, to avoid a dominant effect on the overall processing time.

The above information merges $TA_r DA_r$, the bounding box IoU, and the processing time threshold information into one object function called the error of IoU with area $errIoU_a$. The formula is as follows:

If $(P_t < 34ms)$

$$errIoU_a = IoU \times AAR\left(TA_r, DA_r\right) \tag{13}$$

Else

$$errIoU_a = 1 \tag{14}$$

where P_t is the processing time of color-detection module. Since this is a smaller-the-better problem in the Taguchi method, it is calculated through Equation (2) after the experiment.

V. GRAYSCALE MODULE EXPERIMENT DESIGN

This module indirectly measures how many particle-type contaminants such as dust accumulate through the darkness of the picture. After going through the first two modules, we calculate the average brightness for the rest of the bounding boxes. Then, the module calculates the difference compared with the average brightness of the preliminary picture of the clean exterior wall surface. An example of each process is shown in Figure 10. In this process, when calculating the average brightness, a method of converting brightness information using RGB values contained in pixels is needed. It is necessary to conduct experiments to determine which of the three methods is the best.

A. DESIGN VARIABLES

1) COLOR SPACE (F)

The result depends on which method is used to calculate the average brightness in the area. There are various ways to change the RGB value to a brightness value. The result depends on which color space is to be changed. In this paper, we experimented to select one of three color spaces: YUV, HSV, and CIELab. The Y (luminance) value of the LUV color space is one of the most popular grayscale conversion methods. The V (value) value of the HSV color space represents the brightness value in the color space most commonly used in image editing tools. We used the RGB to HSV conversion formula in this paper [19]. The L (lightness) value of the CIELab color space perceptually represents the brightness value in a uniform color space. We used the RGB to CIELab conversion formula from this paper [20]. The choice of value will affect the output of the module. Therefore, we set this value as a system parameter called design variable F.

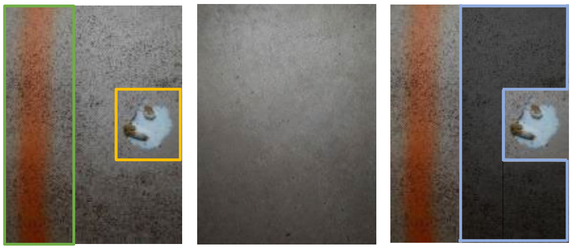


FIGURE 10. Grayscale module's example area and results of each step: (a) image from second module (color-detection module) and bounding boxes from previous modules, (b) reference image and cleaned outer wall with same light condition, and (c) calculating area other than bounding boxes from previous modules.

B. DESIGN OBJECTIVE FUNCTIONS

The information from the grayscale module we want to optimize determines how the average brightness has changed for the reference image. Similar trends must be shown to represent the amount of dust accumulation on the surface with a brightness difference. To measure this, a certain percentage of dust is applied to the specimen, and the similarity of the result of the grayscale module to that ratio is calculated. In the specimen of this paper, the amount of dust was applied to the specimen in different ratios of 1:2:3. Thus, in the ideal case, each measurement is on one straight line. Therefore, after estimating a straight line using linear regression with the Least Square Method (LSM), the R-value (coefficient of determination) is obtained. The closer the R-value is to 1, the better the linear regression. The formula for this is as follows:

$$\bar{x} = \frac{x_1 + x_2 + x_3}{3}, \quad \bar{y} = \frac{y_1 + y_3 + y_3}{3} \tag{15}$$

$$\text{a} = \frac{\sum_{i=1}^{3}(y_i - \bar{y})(x_i - \bar{x})}{\sum(x_i - \bar{x})^2}, \quad b = \bar{y} - a\bar{x} \tag{16}$$

where x_i, y_i isonespecimentdustset (1:2:3),

$x_i = i(i = 1, 2, 3), y_i$: average brightness difference result of the i-th specimen.

Linear regression using LSM is as follows:

$$f_i = ax_i + b \tag{17}$$

The R-value is

$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{18}$$

We calculated the R-value using a specific color space for all specimens. The average of these values is used as a measure of how much the average brightness difference calculated in the color space reflects the actual amount of dust. We defined a new index: the Brightness-to-Dust Correlation ($BTDC$). Its formula is as follows:

$$BTDC_{CS} = \frac{\sum_{j=1}^{N} R_j^2}{N} \tag{19}$$

where R_j^2 is the R-value of j-th specimen dust set (1:2:3),
CS is the current color space calculation method, and
N is the total number of specimen dust sets.

VI. CONDUCTING EXPERIMENTS

We defined the design variable, orthogonal array, and object function for each module above. The entire system proceeds in the order of YOLOv3, color-detection, and grayscale modules. The result of the a module does not affect the previous one. With the value of the found system parameter fixed, the Taguchi method was again used for the next module. In this way, we optimized the entire system.

A. YOLOv3 MODULE

Two design variables of the YOLOv3 module are A (NMS threshold) and B (confidence threshold). The first experiment conducted using the Taguchi method can be found in Appendix 3. When we saw two correlation variable combination results, no strong correlation was found. Therefore, in the first experiment, the value of A did not have much correlation with the result, and it can be seen that B increased the SNR toward the first level.

In the second experiment, the level of A was maintained, whereas B was enlarged to a level below that of the first experiment. The results of the experiment can be found in Appendix 4. The correlation variable combinations still do not exhibit much correlation. Looking only at the design variables, A tended to be highest at levels 1 and 2, and B tended to increase as it fell to level 1. Considering that A and B are the thresholds, if they have the same SNR value, then the one with the higher level must be chosen.

A third experiment was conducted to find the highest A and B values at which the SNR reached the highest point. In the third experiment, the values of levels 1 and 2 of A were slightly expanded. B was enlarged slightly between levels 2 and 3. The results of the experiment can be found in Appendix 5. There was still no correlation in the variables. In the design variable results, A had the highest value at level 2, and B maintained the maximum SNR at level 2. The value at that time was selected as an optimized value. Therefore, the optimum value A was determined to be 0.3, at level 2, and B was 0.25, at level 2. Figure 11 shows the maximum SNR for each conducted experiment.

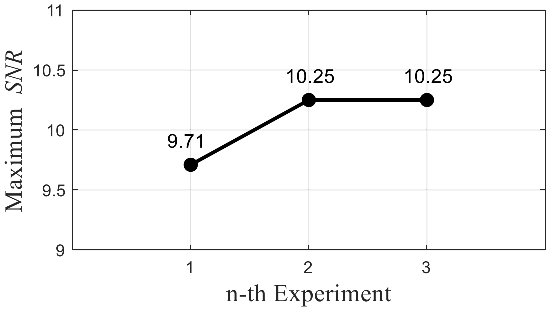


FIGURE 11. Each maximum *SNR* graph of YOLOv3 module optimization.

B. COLOR-DETECTION MODULE

Three design variables of the color-detection module were C (hue margin), D (median filter size), and E (down-sampling size). The first experiment conducted using the Taguchi method can be found in Appendix 6. A weak correlation was found for C∗D and D∗E. C∗E was not considered because of its relatively low correlation. To get a closer look, we drew two variable combination plots and analyzed them. For C∗D combinations, it was best when D and C were at level 1 or when D was at level 3 and C at level 3. However, in both the cases, the direction toward level 3 improved the overall SNR. In the D∗E combination, E at level 2 and D at level 1 were in the better direction.

In the second experiment, C, D, and E were expanded near level 3. The results of the experiment can be found in Appendix 7. There was little correlation between C∗E and C∗D; however, the correlation of D∗E was prominent. We checked the variable combination plot. The SNR improving toward D was level 1, and E was level 1. Because C does not need to consider correlations, we confirmed the design variable result for C; C did not have a considerable amount of influence.

In the third experiment, C was expanded between levels 2 and 3 and D and E were expanded near level 1. The experimental results can be found in Appendix 8. When we confirmed the correlation variable combination result, C∗D and C∗E had a weak correlation. D∗E was found to correlate and confirmed the variable combination plot. The best SNR was achieved when D and E were at level 2. The correlation for C was not noticeable; thus, we confirmed the design variable result of C. Consequently, C exhibited the best SNR at level 2. Therefore, the optimal value was determined as 10 for C at level 2 and 19 for D and E at level 2. Figure 12 shows the maximum SNR for each experiment.

C. GRAYSCALE MODULE

Grayscale modules do not require optimization using the Taguchi method. This is because only one design variable exists in this module. Color space is not a variable that has a continuous value. We did not use the Taguchi method but computed the object function for the three color spaces. We selected the color space that provided the best

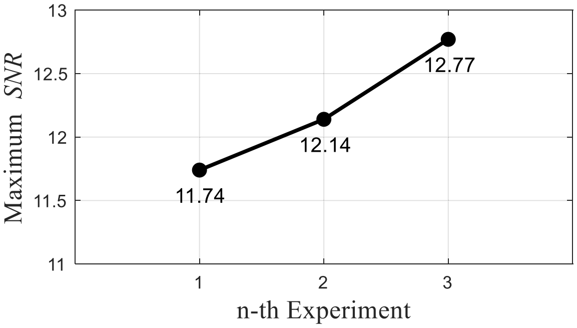


FIGURE 12. Each maximum *SNR* graph of color-detection module optimization.

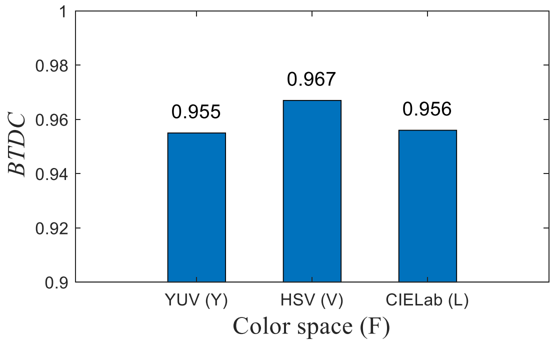


FIGURE 13. Each *BTDC* value of color space design variable.

BTDC value. More detailed experimental results can be found in Appendix 9.

As shown in Figure 13, the method using the V value from HSV delivered the highest *BTDC* value. The method that used the L value from CIELab was good. The color space that delivered the lowest *BTDC* value was the method using the L value from LUV. Therefore, we decided to use the HSV color space as the RGB-to-grayscale conversion method.

VII. INTEGRATED TEST & RESULTS

The optimized system parameters were applied to the integrated detection system. The 144 specimens introduced in Section 2D were used for the test, and the *SNR*s of the YOLOv3 module and color-detection module were then calculated. The change was compared using the initial *SNR* as the highest *SNR* in the first experiment of each module.

TABLE 1. Initial and optimized values of design variables.

System Parameter	YOLOv3 module		Color-detection module		
	NMS Threshold	Confidence Threshold	Hue margin	Median filter size	Down sampling size
Initial value	0.3	0.3	9	23	15
Optimized value	0.3	0.25	10	19	19

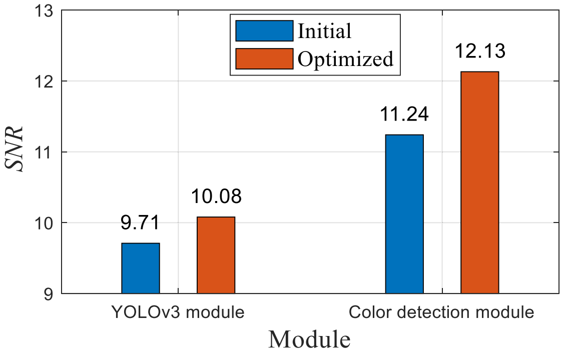


FIGURE 14. *SNR* differences between initial system parameters and optimized parameters.

TABLE 2. Design variables.

Design Variable	A	B
System Parameter	NMS Threshold	Confidence Threshold

TABLE 3. Orthogonal array.

Row	A	B	AB(1)	AB(2)
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

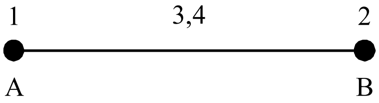


FIGURE 15. Linear graph.

TABLE 4. Design variables.

Design Variable	C	D	E
System Parameter	Hue margin	Median Filter Size	Down-sampling Size

The value of each system parameter can be checked in Table 1. The increase in *SNR* of each module can be noted in Figure 14.

VIII. CONCLUSION

This paper briefly introduced a machine-vision system that detects contaminants on building facades and optimized

TABLE 5. Orthogonal array.

Row	C	D	C·D (1)	C·D (2)	E	D·E (1)	D·E (2)	C·E (1)	C·E (2)
1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2
3	1	1	1	1	3	3	3	3	3
4	1	2	2	2	1	1	1	2	3
5	1	2	2	2	2	2	2	3	1
6	1	2	2	2	3	3	3	1	2
7	1	3	3	3	1	1	1	3	2
8	1	3	3	3	2	2	2	1	3
9	1	3	3	3	3	3	3	2	1
10	2	1	2	3	1	2	3	1	1
11	2	1	2	3	2	3	1	2	2
12	2	1	2	3	3	1	2	3	3
13	2	2	3	1	1	2	3	2	3
14	2	2	3	1	2	3	1	3	1
15	2	2	3	1	3	1	2	1	2
16	2	3	1	2	1	2	3	3	2
17	2	3	1	2	2	3	1	1	3
18	2	3	1	2	3	1	2	2	1
19	3	1	3	2	1	3	2	1	1
20	3	1	3	2	2	1	3	2	2
21	3	1	3	2	3	2	1	3	3
22	3	2	1	3	1	3	2	2	3
23	3	2	1	3	2	1	3	3	1
24	3	2	1	3	3	2	1	1	2
25	3	3	2	1	1	3	2	3	2
26	3	3	2	1	2	1	3	1	3
27	3	3	2	1	3	2	1	2	1

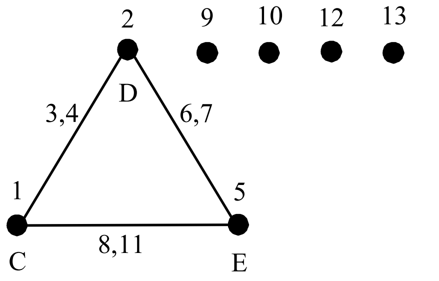


FIGURE 16. Linear graph.

TABLE 6. Design variable level.

Level	1	2	3
A (NMS Threshold)	0.3	0.5	0.7
B (Confidence Threshold)	0.3	0.5	0.7

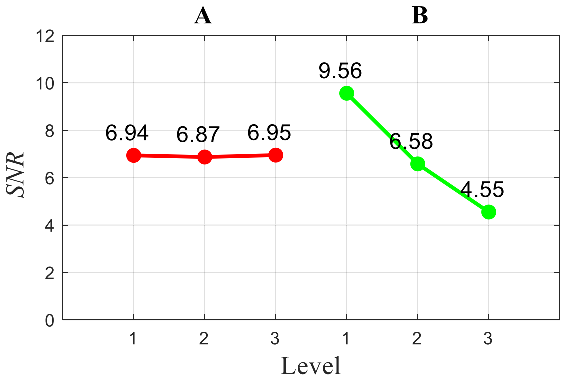


FIGURE 17. Design variable results.

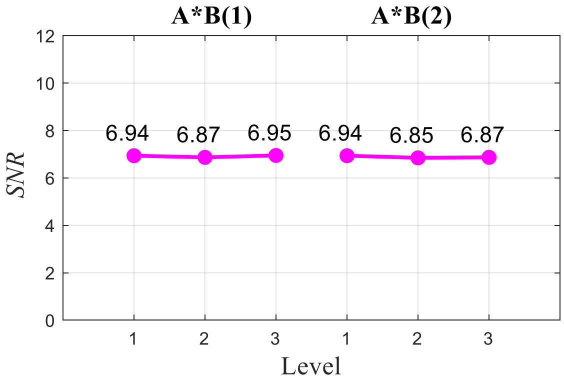


FIGURE 18. Correlation variable combination results.

TABLE 7. Design variable level.

Level	1	2	3
A (NMS Threshold)	0.3	0.5	0.7
B (Confidence Threshold)	0.1	0.2	0.3

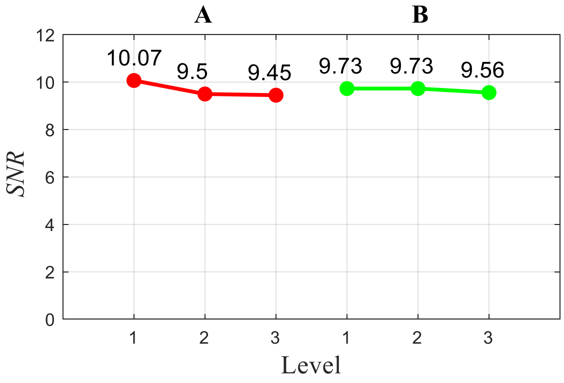


FIGURE 19. Design variable results.

the system using the Taguchi method. Camera height and image brightness were selected as user conditions, and 144 specimen images were prepared accordingly. The images were captured at distinct brightness levels, rotation angles, and facade distances. Optimization was performed using the Taguchi method in the following order: the YOLOv3 module, color-detection module, and grayscale

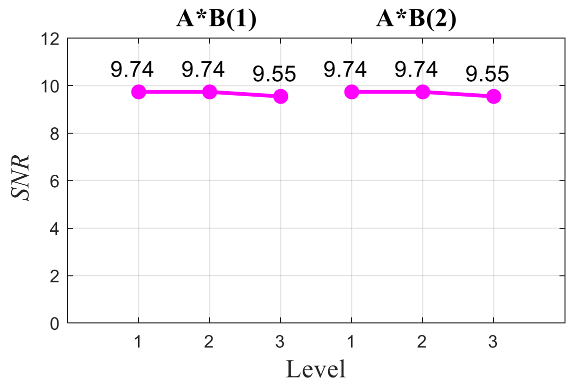


FIGURE 20. Correlation variable combination results.

TABLE 8. Design variable results.

Level	1	2	3
A (NMS Threshold)	0.2	0.3	0.4
B (Confidence Threshold)	0.2	0.25	0.3

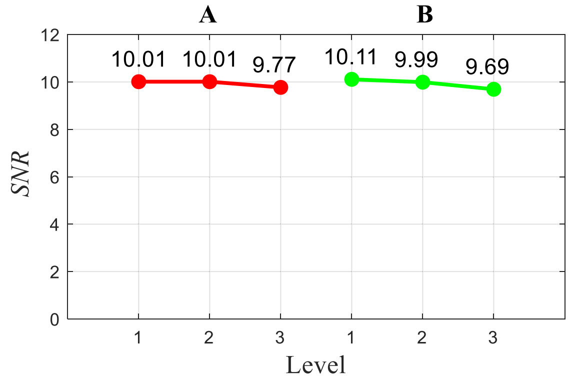


FIGURE 21. Design variable results.

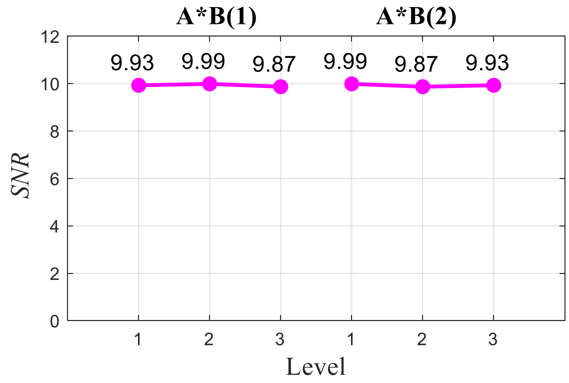


FIGURE 22. Correlation variable combination results.

module. The *SNR* difference between the initial and optimized values highlighted the improvement in overall system performance.

TABLE 9. Design variable level.

Level	1	2	3
C (Hue margin)	1	5	9
D (Median filter size)	7	15	23
E (Down-sampling size)	7	15	23

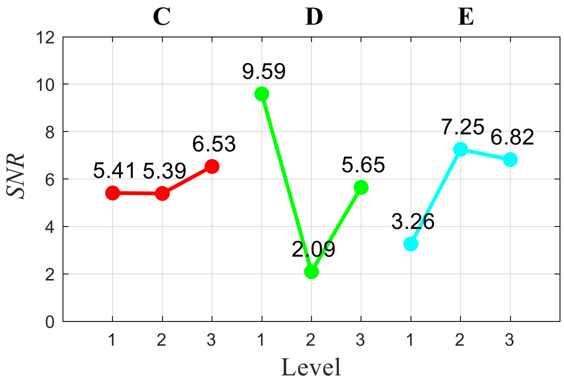


FIGURE 23. Design variable results.

However, some problems were also encountered. First, in the optimization process of the grayscale module, the ratio of amout of dust applied to the surface was only 1:2:3. Current optimization result would be less stable than the experiments in more steps of ratio with amount of dust. Next, when optimizing the color-detection module, traces of avian feces were found on the rusty stain. If the YOLOv3 module detects avian feces in the previous module and wraps it in a box, then the next module does not check that area. This gives rise to errors if the box contains rusty stains.

In the future studies, we shall investigate the use of object segmenting instead of using the bounding box. This shall allow us to separate contaminants of different types more effectively. In addition, we shall consider further optimization of the grayscale module by producing more diverse specimens with a dust ratio of 5:7 or more steps.

We plan to perform the field test. The experiments in this paper is done in lab condition; therefore, there are some factors to be considered to be check in the field test. The illumination problem due to the high sunlight is the most sensitive factor to be considered. The deviation of the lights in between day and night should be compensated carefully by using hardware design and software filtering. The problems from integration with cleaning device such as moving speed and vibration of the camera should also be considered to determine the robust design factors for the contamination detection. We are going to be share the experience on the field test after performing and analyzing the results shortly.

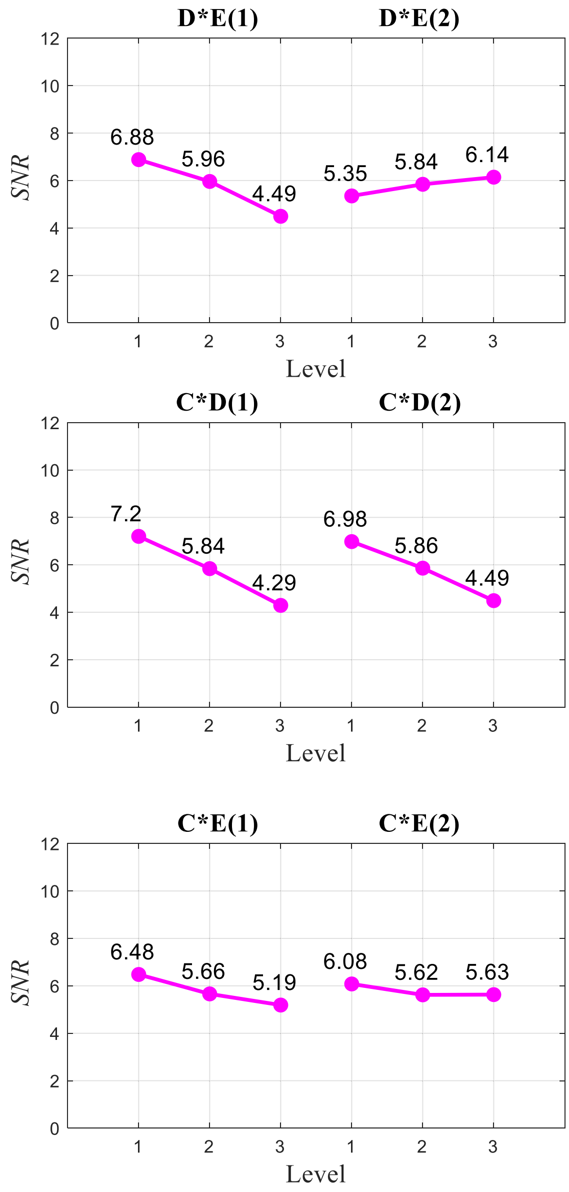


FIGURE 24. Correlation variable combination results.

APPENDIX

A. YOLOv3 MODULE EXPERIMENT PLAN

See Tables 2 and 3 and Figure 15.

B. EXPERIMENT PLAN FOR COLOR-DETECTION MODULE

See Tables 4 and 5 and Figure 16.

C. YOLOv3 MODULE 1ST OPTIMIZATION RESULTS

See Table 6 and Figures 17 and 18.

D. YOLOv3 MODULE 2nd OPTIMIZATION RESULTS

See Table 7 and Figures 19 and 20.

E. YOLOv3 MODULE 3rd OPTIMIZATION RESULTS

See Table 8 and Figures 21 and 22.

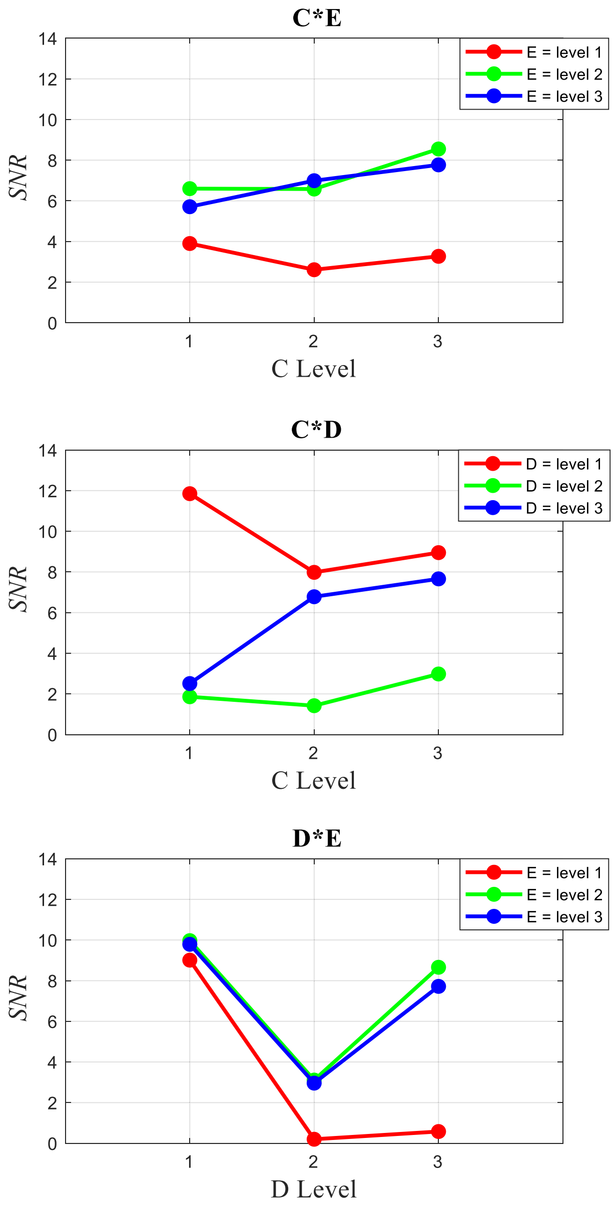


FIGURE 25. Variable combination plot.

TABLE 10. Design variable level.

Level	1	2	3
C (Hue margin)	1	5	9
D (Median filter size)	19	23	27
E (Down-sampling size)	19	23	27

F. COLOR-DETECTION MODULE 1st OPTIMIZATION RESULTS

See Table 9 and Figures 23–25.

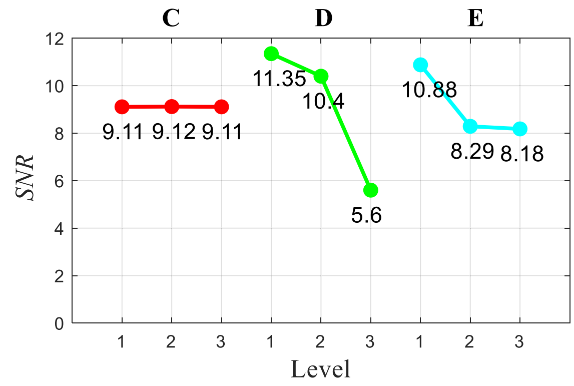


FIGURE 26. Design variable results.

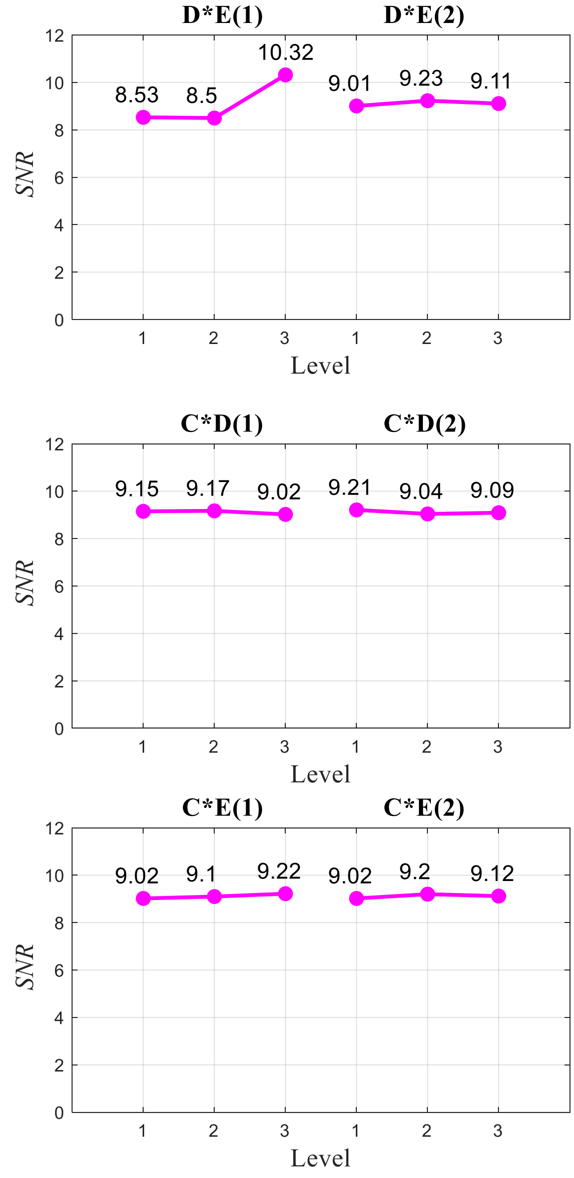


FIGURE 27. Correlation variable combination results.

G. COLOR-DETECTION MODULE 2nd OPTIMIZATION RESULTS

See Table 10 and Figures 26–28.

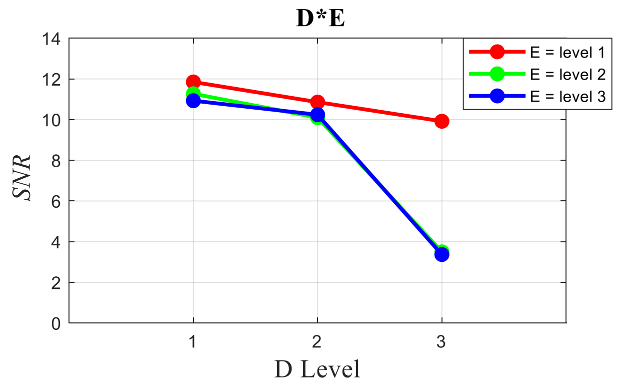


FIGURE 28. Variable combination plot.

TABLE 11. Design variable level.

Level	1	2	3
C (Hue margin)	9	10	11
D (Median filter size)	17	19	21
E (Down-sampling size)	17	19	21

TABLE 12. Grayscale module optimization results.

Height (Level)	Brightness (Level)	R-value		
		YUV	HSV	CIELAB
1	1	0.918	0.955	0.925
1	2	0.983	0.999	0.987
1	3	0.994	0.996	0.994
2	1	0.929	0.950	0.925
2	2	0.920	0.919	0.919
2	3	0.986	0.984	0.986
Average		0.955	0.967	0.956

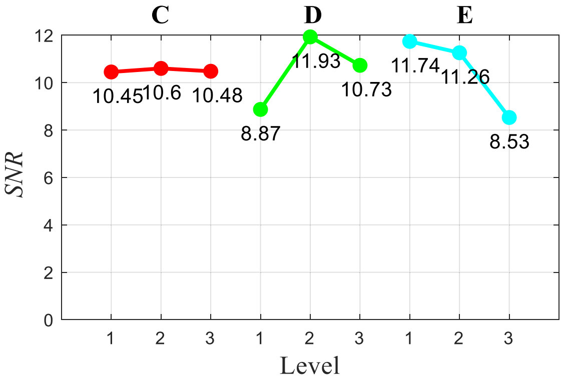


FIGURE 29. Design variable results.

H. COLOR-DETECTION MODULE 3rd OPTIMIZATION RESULTS

See Table 11 and Figures 29–31.

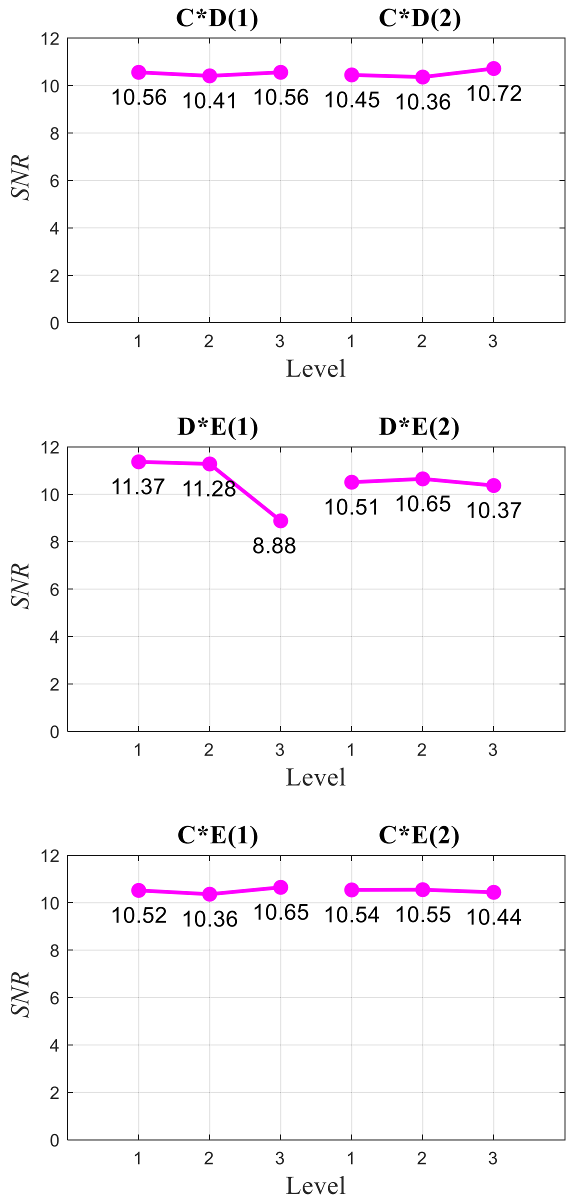


FIGURE 30. Correlation variable combination results.

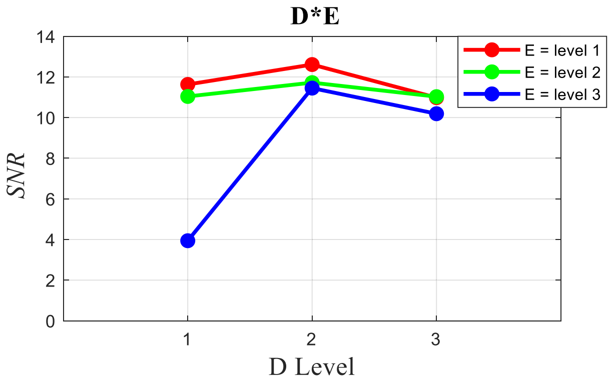


FIGURE 31. Variable combination plot.

I. GRAYSCALE MODULE OPTIMIZATION RESULTS

See Table 12.

REFERENCES

[1] The Skyscraper Center. *The Global Tall Building Database of the CTBUH*. Accessed: Feb. 10, 2019. [Online]. Available: http://www.skyscrapercenter.com/countries

[2] T. Seo, Y. Jeon, C. Park, and J. Kim, "Survey on glass and Façade-cleaning robots: Climbing mechanisms, cleaning methods, and applications," *Int. J. Precis. Eng. Manuf.-Green Tech.*, vol. 6, no. 2, pp. 367–376, Apr. 2019.

[3] J. Hong, G. Park, J. Lee, J. Kim, H. S. Kim, and T. Seo, "Performance comparison of adaptive mechanisms of cleaning module to overcome step–shaped obstacles on Façades," *IEEE Access*, vol. 7, pp. 159879–159887, 2019.

[4] S. Yoo, I. Joo, J. Hong, C. Park, J. Kim, H. S. Kim, and T. Seo, "Unmanned high–rise Façade cleaning robot implemented on a gondola: Field test on 000-building in Korea," *IEEE Access*, vol. 7, pp. 30174–30184, 2019.

[5] J. Hong, S. Yoo, I. Joo, J. Kim, H. S. Kim, and T. Seo, "Optimal parameter design of a cleaning device for vertical glass surfaces," *Int. J. Precis. Eng. Manuf.*, vol. 20, no. 2, pp. 233–241, Feb. 2019.

[6] I. Joo, J. Hong, S. Yoo, J. Kim, H. S. Kim, and T. Seo, "Parallel 2-DoF manipulator for wall-cleaning applications," *Autom. Construct.*, vol. 101, pp. 209–217, May 2019.

[7] R. Harig, R. Braun, C. Dyer, C. Howle, and B. Truscott, "Short-range remote detection of liquid surface contamination by active imaging Fourier transform spectrometry," *Opt. Express*, vol. 16, no. 8, p. 5708, Apr. 2008.

[8] A. J. Sedlacek, III, N. S. Higdon, D. A. Richter, and M. D. Ray, "Short-range noncontact detection of surface contamination using Raman lidar," *Proc. SIPE*, vol. 4577, pp. 95–104, Feb. 2002.

[9] J. Lee, G. Park, J. Hong, H. S. Kim, and T. Seo, "Contamination detection and classification for an automated façade cleaning operation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Macau, China, Nov. 2019, p. 845.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real–time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[11] A. R. Smith, "Color gamut transform pairs," *SIGGRAPH Comput. Graph.*, vol. 12, no. 3, pp. 12–19, Aug. 1978.

[12] G. Taguchi, *Taguchi on Robust Technology Development: Bringing Quality Engineering Upstream*. New York, NY, USA: ASME Press, 1993.

[13] G. S. Peace, *Taguchi Methods: A Hands-On Approach*. Reading, MA, USA: Addison-Wesley, 1993.

[14] J. T. Krishakant, B. Mohit, and R. Kumar, "Application of Taguchi method for optimizing turning process by the effects of machining parameters," *Int. J. Eng. Adv. Technol.*, vol. 2, no. 1, pp. 263–274, Oct. 2012.

[15] J. A. Ghani, L. A. Choudhury, and H. H. Hassan, "Application of Taguchi method in the optimization of end milling," *J. Mater. Process, Tech.*, vol. 145, no. 1, pp. 84–92, 2004.

[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[17] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[18] R. Srinivas and S. Panda, "Performance analysis of various filters for image noise removal in different noise environment," *Int. J. Adv. Comput. Res.*, vol. 3, pp. 47–52, Dec. 2013.

[19] M. Podpora, G. P. Korbas, and A. Kawala-Janik, "YUV vs. RGB: Choosing a colourspace for human-machine interaction," in *Proc. Federated Conf. Comput. Sci. Inform. Syst.*, vol. 3, 2014, pp. 29–34.

[20] C. Connolly and T. Fleiss, "A study of efficiency and accuracy in the transformation from RGB to CIELAB color space," *IEEE Trans. Image Process.*, vol. 6, no. 7, pp. 1046–1048, Jul. 1997.

JISEOK LEE received the B.S. degree in mechanical engineering from Hanyang University, in 2019, where he is currently pursuing the M.S. degree in mechanical engineering. His research interests lie in the areas of robot automatic control.

GARAM PARK received the B.S. degree in mechanical engineering from Hanyang University, in 2019, where he is currently pursuing the M.S. degree in mechanical engineering. His research interests lie in the areas of robot mechanism design.

YECHEOL MOON received the B.S. degree from the School of Mechanical Engineering, Hanyang University, Seoul, South Korea, in 2018, where he is currently pursuing the M.S. degree with the Department of Mechanical Convergence Engineering. His research interests include the design and control of robotic platforms and underwater robots.

SUNGON LEE received the B.S. degree in mechanical design and product engineering from Seoul National University, Seoul, South Korea, in 1997, the M.S. degree in mechanical engineering from POSTECH, Pohang, South Korea, in 1999, and the Ph.D. degree in mechano-informatics from the University of Tokyo, Tokyo, Japan, in 2008. From 2010 to 2012, he was a Postdoctoral Research Fellow with the Center for Systems Biology, Massachusetts General Hospital, Harvard Medical School, Boston, MA, USA. From 1999 to 2015, he was a Senior Research Scientist with the Korea Institute of Science and Technology, Seoul. Since 2015, he has been with the School of Electrical Engineering, Hanyang University, Ansan, South Korea. He is currently an Associate Professor. His research interests include biomedical robotics, surgical robots, medical image processing, and motion compensation systems.

TAEWON SEO (Member, IEEE) received the B.S. and Ph.D. degrees from the School of Mechanical and Aerospace Engineering, Seoul National University, South Korea. He was a Postdoctoral Researcher with the Nanorobotics Laboratory, Carnegie Mellon University, a Visiting Professor with the Biomimetic Millisystems Laboratory, UC Berkeley, and an Assistant Professor with the School of Mechanical Engineering, Yeungnam University, South Korea. He is currently an Associate Professor with the School of Mechanical Engineering, Hanyang University, South Korea. His research interest includes robot design, analysis, control, optimization, and planning. He received the Best Paper Award of the IEEE/ASME Transaction on Mechatronics, in 2014, and currently working as a Technical Editor of IEEE/ASME Transactions on Mechatronics, an Associate Editor of the IEEE Robotics and Automation Letters and *Intelligent Service Robots*.