Slag Removal Path Estimation by Slag Distribution and Deep Learning

Junesuk Lee[1][a], Geon-Tae Ahn[2], Byoung-Ju Yun[1][b] and Soon-Yong Park[1][c]

[1]*School of Electronics Engineering, Kyungpook National University, Daegu, South Korea*

[2]*Research Institute of Industrial Science and Technology, Pohang, South Korea*

Keywords: Path Estimation, Deep Learning, Intelligent Robots, Industrial Robots.

Abstract: In the steel manufacturing process, de-slagging machine is used to remove slag floating on molten metal in a ladle. In general, temperature of floating slag on the surface of the molten metal is above 1,500℃. The process of removing such slag at high temperatures is dangerous and is only performed by trained human operators. In this paper, we propose a deep learning method for estimating the slag removal path to automate slag removal task. We propose an idea of developing a *slag distribution image structure(SDIS)*; combined with a deep learning model to estimate the removal path in an environment in which the flow of molten metal cannot be controlled. The *SDIS* is given as the input into to the proposed deep learning model, which we train by imitating the removal task of experienced operators. We use both quantitative and qualitative analyses to evaluate the accuracy of the proposed method with the experienced operators.

1 INTRODUCTION

Recently, in the field of intelligent robotics, artificial intelligence technology has significant research topics (Kim. J. et al., 2018), (Cauli et al., 2018). An intelligent robot is a robot that can recognize the external environment and judge the situation by itself to operate autonomously. Intelligent robots are mainly divided into industrial robots and service robots. Industrial robots perform dangerous, hazardous, or simple repetitive tasks on behalf of humans. These robots offer significant benefits in the manufacturing industry: labor cost, productivity, and quality. In particular, industrial robots improve the working environment for humans. In industrial environments, people are always threatened by polluted air, dangerous materials, and so on. Introducing robots into industrial environments can consequently reduce the severity of most dangerous effects to humans.

In steel company, various by-products/residues, such as slag, dust, and sludge are produced when steel is produced. Molten iron is located in the lower part of the furnace, and by-products drift on it. To produce high quality steel, such residue must be removed from the furnace. In general steel company, slag removal is performed using a skimmer as shown in Figure 1.

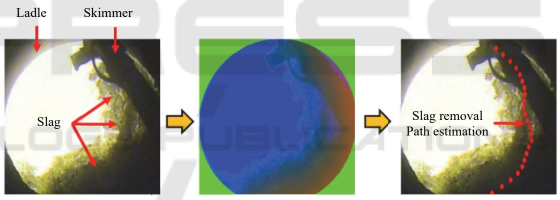


Figure 1: System overview.

Since steel production has dangerous working environment, there is a high probability of accidents involving human operators. This is because the molten metal has a temperature of about 1,500℃ or higher and the view of the metal is covered by some dust. To protect operators getting exposed to such dangerous conditions, automated robots capable of performing human tasks are required.

In this paper, we propose an accurate slag removal path estimation method which can easily be embedded into any automated system. The system overview of the proposed method is briefly shown in

[a] https://orcid.org/0000-0002-1702-6720
[b] https://orcid.org/0000-0002-9898-2262
[c] https://orcid.org/0000-0001-5090-9667

Lee, J., Ahn, G., Yun, B. and Park, S.
Slag Removal Path Estimation by Slag Distribution and Deep Learning.
DOI: 10.5220/0008944602460252
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020) - Volume 4: VISAPP*, pages 246-252
ISBN: 978-989-758-402-2; ISSN: 2184-4321

Figure 1, where the skimmer moves from top to bottom along the ladle. The overview of the proposed system has two distinctive stages: an image transformation stage which contains slag distribution information, and a slag removal path estimation stage based on deep learning. The main contributions of this paper are three-fold:

1. propose a learning model that mimics the slag removal path of a skilled human operators to estimate its removal path.
2. create a slag distribution image structure that contains slag distribution information from color images.
3. estimate the slag removal path in real-time, and the experimental results verify the performance of the proposed method.

The structure of this paper is organized as follows: First, the related work is discussed in Section 2. Section 3 describes the creation of the proposed slag distribution image and how training data can be extracted. Section 4 describes the structure and post-processing of the deep learning model used to predict the slag removal path. Section 5 discusses about the working environment and quantitative and qualitative accuracy analyses through a few experimental results. Finally, we conclude in Section 6.

2 RELATED WORK

In this section, we briefly discuss previous studies which are related to automated robots of desk cleaning, rock excavation, slag removal, and route prediction. (Kim. J. et al., 2018) proposed a desk cleaning technique using the iCub humanoid Robot for cleaning graffiti and lentils from a desk. For a robot to clean the top of a desk automatically, it must recognize the material on the desk and estimate the path to clean it. In this study, a human instructor teaches the robot how to perform cleaning tasks. Task Parametrized Gaussian Mixture Model (TP-GMM) is used to encode the demo variables and to properly generalize the features. However, the parameterization of TP-GMM is very difficult because it requires partitioning and extracting complex images of small tables. Therefore, while the instructor demonstrates the cleaning task, a trained deep neural network is used to extract parameters from the robot camera image.

(Fukui et al., 2015) discussed about an Automated Ore Excavator. To carry out autonomous excavation of rocks, it is necessary to recognize the state of the fragmented rock piles and plan the appropriate excavation operation accordingly. They proposed an imitation-based motion planning method and developed a rock pile condition recognizer with an excavation motion planner. To verify the proposed method, they developed a 1/10 scale excavation model and conducted excavation experiments. Experimental results showed that rock piles could be distinguished according to surface shape and particle distribution, where the number and the variety of training data proved important for realizing high productivity excavation.

(Kim. J. S. et al., 2018) conducted a study to remove slag using a de-slagging machine. In general, de-slagging machines can only be controlled by trained professionals. In their research, they proposed a method for estimating the slag removal path automatically using CNN. They trained their network by extracting block regions based on the actions of an experienced specialist. They performed backtracking and curve fitting to properly estimate the removal path and compared with the path of the experienced expert.

(Minoura et al., 2018) proposed a path prediction method that takes target object attributes and physical environment information into account. Previous path prediction methods using deep learning architecture took into account the physical environment of a single target, such as a pedestrian. However, they proposed a route prediction method that could consider multiple target types. The method represents the attributes as one-hot vectors and encodes the physical attributes through convolutional layers. Furthermore, we used relative coordinates as the past motion history of prediction targets. They verified the proposed method using the Stanford drone dataset.

3 TRAINING DATA ACQUISITION

3.1 Slag Distribution Image Generation

In slag removal task, an area with high slag distribution is removed first. The reason is that high slag distribution means that the slag is concentrated in the area. By removing dense slag areas, it results in efficient removal task.

In this section, we discuss the design architecture of the slag distribution image which is proposed to train our deep learning network (Figure 2). It consists of 3 channels: grayscale, morphology, and distance transform images. In addition, it also contains information that can be used to distinguish the inside and outside of the ladle. In our method, we determine that the slag exists only inside the inner part of the ladle and the skimmer is moved only within this area. We also generate a binary ladle image, which consists of zero and one as shown in Figure 3. The inside of the ladle is labeled as 1 and the outside is 0.

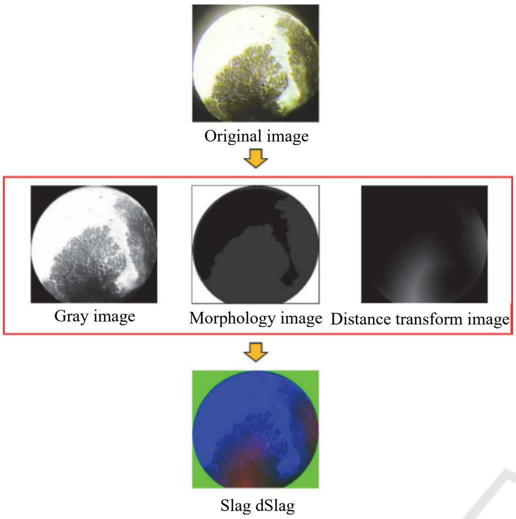


Figure 2: Slag distribution image.

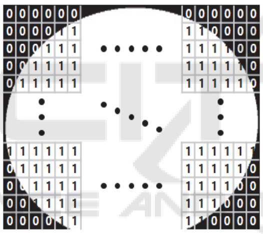


Figure 3: binary ladle image.

Figure 4 shows how the slag distribution image is created and how it is converted from an input color image. First, we convert the input RGB image into grayscale (Figure 4(a)) and perform the Hadamard multiplication using the binary ladle image. This operation results in zero value outside the ladle area in the converted grayscale image (Figure 4(b)). The operator '∘' in Figure 4 represents the Hadamard multiplication operation.

After getting the inverse binary image from the converted grayscale image, we apply the opening morphology operation (erosion and dilation) as shown in Figure 4(d). We used a 3x3 kernel for erosion and a 9x9 kernel for dilation, respectively. This morphology operation not only removes small slag chunks but also fills holes in large chunks. Next, we perform the Hadamard multiplication on Figure 4(d) with the binary ladle image. Next, we multiply the figure 4(e) by 0.2 scale and add the output of the function G(ladle binary image). The final result of the morphology operation is shown in Figure 4(f). Function $G(x)$ is the same as Equation 1. Operation symbol $\oplus$ means bit XOR operation.

$$G(x) = \overline{(x \oplus x)} * 255 \quad (1)$$

To generate the distance transform image, we first apply the distance transform algorithm (Felzenszwalb and Huttenlocher, 2012) on Figure 4(d). The Hadamard multiplication is applied on Figure 4(g) to distinguish between inside and outside of the ladle. Finally, we merge the grayscale, morphology, and distance transform images to create the slag distribution image as shown in Figure 4(i). The three

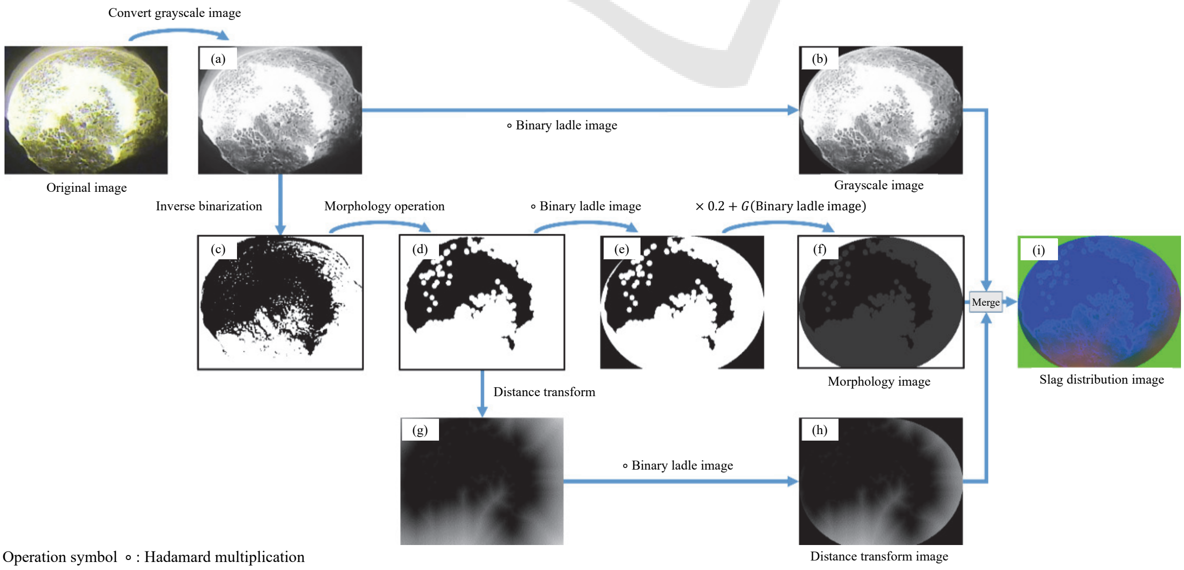


Figure 4: Pipeline for constructing the proposed slag distribution image.

separate channels in this slag distribution image resemble the BGR channels in a general color image.

3.2 Proposed Training Data Format and Data Acquisition Method

In this paper, we use a learning-based algorithm to solve the slag removal path estimation problem. In the proposed method, we equally divide the height (y-coordinate) of the slag distribution image into 24 fixed intervals as shown in Figure 5. We perform the learning and estimating processes only along the x-coordinate. Fixing the y-coordinate reduced the total computational cost and increased the efficiency of the path estimation problem.

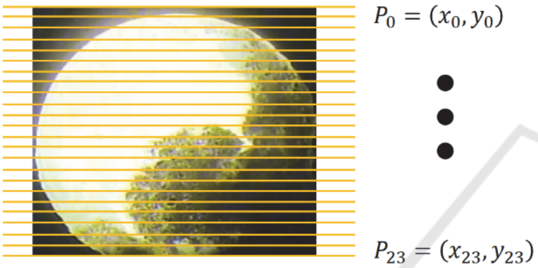
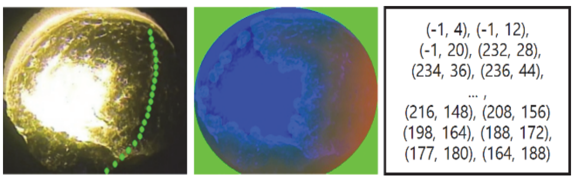


Figure 5: Proposed training data format.

To train our network, we collect training data by imitating the removal path of a skilled human operator. An example of a collected dataset is shown in Figure 6. The left side shows a recorded removal path, the middle is the slag distribution image, and the right is a 2D slag removal path-vector that stores the coordinates of the recorded control points. We use the slag distribution image and the 2D path vector to train our deep learning model. After training our network, we use only the slag distribution image in the test phase. Some examples of manual path recording are shown in Figure 7, which will be used in later sections.



(a) (b) (c)

Figure 6: Training data, (a) Removal path visualization image (b) Training image (c) Removal path vector.

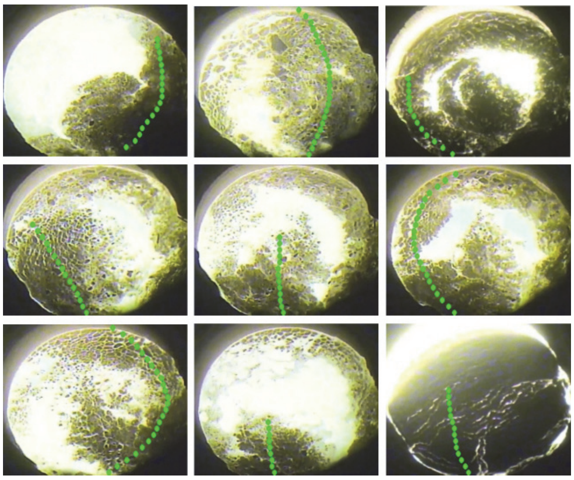


Figure 7: Acquisition of human operator's removal path.

The slag removal path-vector $P(x, y)$ can be defined mathematically as in Equation (2). In this equation, w and h represent the width and height of the training image. x, y, and i are of an integer data type. If slag removal is not need in the i-th vector position, we set its x-coordinate to -1. The green points in Figure 7 are displayed only if this condition is satisfied.

$$P(x,y) = \left\{ (x,y) \;\middle|\; \begin{array}{c} 0 \le x < w, \\ y = \frac{h}{24} * i, \\ 0 \le i < 24 \end{array} \right\} \qquad (2)$$

4 SLAG REMOVAL PATH ESTIMATION

4.1 Deep Network Architecture

The network structure proposed in this research is shown in Figure 8. We use the ResNet50 model's Skip connection (He et al., 2016) to minimize gradient vanishing problem. When training the network, we use both the slag distribution image and x-coordinates of the 2D vector as the inputs. Our network model outputs the respective x-coordinate value corresponding to each y-coordinate. The input image size is 288x192x3, and the output of the trained network model is 24x1x1, which means the 24 x-coordinate values. We use a total of 25 layers to create the proposed network. All convolution layers in the proposed network use elu-activation (Clevert et al., 2015) and zero-padding. A key point of the proposed network is that the filter size of all convolution is 5x5. In general, the object recognition problem using deep learning uses a convolution filter size of 3x3.

However, since it is important to recognize the distribution of slag, we use a large filter size to observe a larger area.

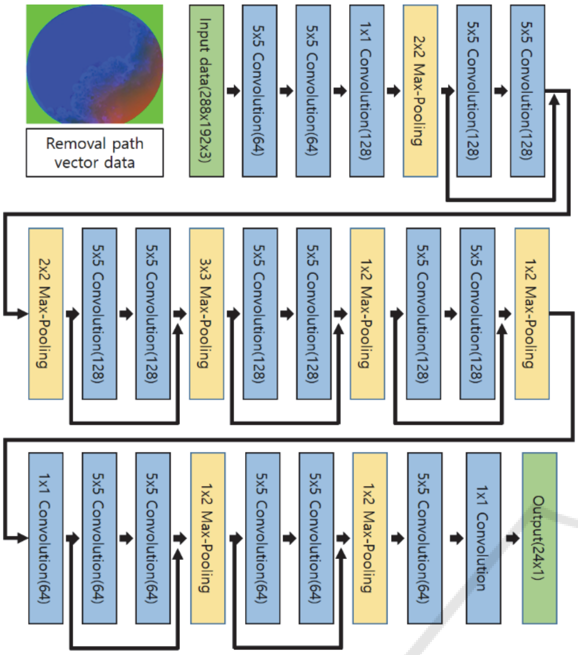


Figure 8: Proposed network model for slag removal path estimation.

4.2 Loss Function

In this paper, we use the loss function as shown in Equation (3) to train our network. This loss function consists of two loss-terms. λ is a constant parameter for fine-tuning the weight for each term. In this paper, λ_{data} is set to 10 and $\lambda_{laplacian}$ is set to 1.

$$loss_{removal\_path} = \lambda_{data} loss_{data} + \lambda_{laplacian} loss_{laplacian} \tag{3}$$

$$loss_{data} = \sum_{i=0}^{M} \|v_i(t_i - x_i)\|_2^2 \tag{4}$$

$$v_i = \begin{cases} 1, & if \quad t_i \neq -1 \\ 0, & else \end{cases} \tag{5}$$

$loss_{data}$ is an expression for finding weight values optimized for training data. In $loss_{data}$, M means the number of output elements. The proposed network outputs 24 x-coordinate values, so we set M to 23. t_i means the answer label. x_i means the output of the network. v_i is a variable that excludes the control point whose x coordinate point is -1 on the answer label. v_i is defined as in Equation (5).

$$loss_{laplacian} = \sum_{i=1}^{M-1} \|x_{i-1} - 2 * x_i + x_{i+1}\|_2^2 \tag{6}$$

$loss_{laplacian}$ is defined by Equation (6). $loss_{laplacian}$ is an expression that minimizes the difference in magnitude between the values of listed x-coordinates. In other words, it is a formula that minimizes the curvature of the slag removal path. This equation is necessary because the curvature of the slag removal path is large when a human operator removes slag. That is, the formula is to make an output value x_i that minimizes the difference between x_i and x_{i-1} and the difference between x_i and x_{i+1}.

4.3 Path Estimation

The proposed slag removal path estimation model yields 24 values for the x-coordinate. These output values correspond to the 24-equally spaced y-coordinate values, as described in Section 3.2. As shown in Figure 7, the number of control points are sometimes 24 or less depending on the distribution of slag. To decide the removal path along only in the image areas covered by slag, output points in the de-slag area should be removed. In this paper, such inefficient slag removal path coordinates are excluded as shown in Figure 9. The exclusion steps of inefficient removal path coordinates are as follows:

1) In the grayscale channel of the slag distribution image, we search the intensity values of control points in the order from the top ($k = 0$) to bottom ($k = 23$).
2) If the intensity of a control point is less than γ, this area is considered to be the area where slags exist. γ is the intensity threshold value for the slag area. We define h as the lowest index number among the control points where the slag exists.
3) We extract the h th to k th as the final slag removal path.

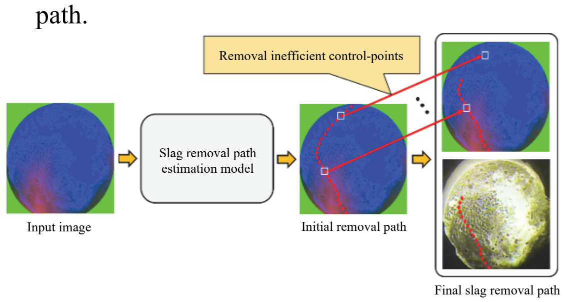


Figure 9: Exclusion of unnecessary control points.

5 EXPRIMENTAL RESULTS

We have done experiments using many ladle images to verify the performance of the proposed slag removal path estimation method. When performing these experiments, we used Windows-10 (64 bit) and an NVIDIA Titan XP graphics card. Also, our implementation uses the Keras-GPU library. To train the proposed network model for slag removal path estimation, we use a large number of training data as shown in Table 1. We assign a high weight to the training data count as there is not enough dataset. We use the Adadelta optimizer method (Zeiler, 2012) to train the proposed network model. We set the batch size to 100 and the learning rate to 0.8 in the learning options.

Table 1: Number of Training, Validation and Test set.

Training set	Validation set	Test set
1,110	139	139

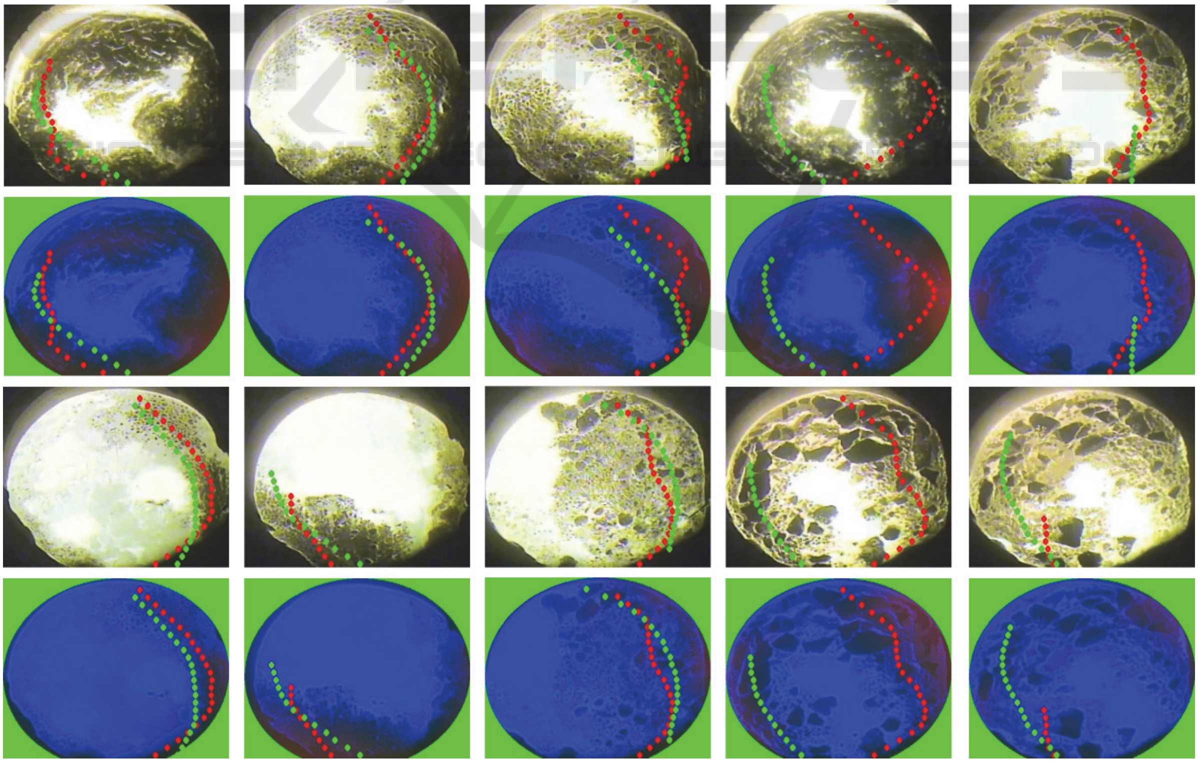
Figure 10 is a comparison between the estimated removal paths from the proposed method with those of experienced operators, which we define as the GT (Ground Truth).

The experimental results are analysed by both qualitative and quantitative ways. First, qualitative analysis is as follows. In Figure 10, (a), (b), and (c) estimated slag removal paths look similar to GT. However, the result in Figure 10(d) differs from the GT, it estimates the best efficient path. Figure 10(e) shows bad performance compared with GT.

As quantitative analyses, we evaluate the amount of slag removal in the image. We compare the estimated removal path with the amount of slag removed from the GT. Slag removal is measured by the amount of slag inside the 3x3 kernel at the all the control points. The measurement method counts the intensity value of less than γ in the grayscale channel of the slag distribution image. We measure the slag removal of 89 images and compare the proposed method using Equation (7). The average % of slag removal of the proposed method is about 77.33% compared with GT.

$$S_{ratio} = \frac{The\ amount\ of\ Proposed\ mothod}{The\ amount\ of\ GT} \times 100 \tag{7}$$

The processing time of the proposed method is shown in Table 2. The conversion time from the input image to the generation of the proposed slag distribution image takes 28ms. Slag removal path



(a) (b) (c) (d) (e)

Figure 10: Slag removal path estimation results of various ladle images: Green dots are the result of an experienced professional (Ground Truth), RED dots are the estimated result of the proposed method.

estimation takes 9ms to process. Therefore, the total average processing time is 37ms.

Table 2: Processing time.

Convert to Slag distribution image	Slag removal path estimation
28ms (35.71 fps)	9ms (111.11 fps)

6 CONCLUSIONS

In this paper, we propose an efficient slag removal path estimation method based on a deep learning network. We introduce a slag distribution image structure, which includes 3-channel slag distribution information. This image is used as the input of the network, which was trained by slag removal path information by experienced human operators. We obtain optimal slag removal path information through the outputs of the network, and apply post-processing techniques to remove invalid control points from the output of the trained model. In experiments, we visualize that the estimated slag removal path has an average of 77.33% performance enhancement compared to the manually recorded path. The total average processing time is about 37ms, ensuring its real-time capabilities.

ACKNOWLEDGEMENTS

This work was supported by 'The Cross-Ministry Giga KOREA Project' grant funded by the Korea government(MSIT) (No.GK17P0300, Real-time 4D reconstruction of dynamic objects for ultra-realistic service)
This research was supported by the National Research Foundation of Korea (NRF) funded by the Korea government (MSIT: Ministry of Science and ICT) ((No. 2018M2A8A5083266))

REFERENCES

Kim, J., Cauli, N., Vicente, P., Damas, B., Cavallo, F. and Santos-Victor, J., 2018, April. “iCub, clean the table!” A robot learning from demonstration approach using Deep Neural Networks. In 2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pages. 3-9. IEEE.

Cauli, N., Vicente, P., Kim, J., Damas, B., Bernardino, A., Cavallo, F. and Santos-Victor, J., 2018. Autonomous table-cleaning from kinesthetic demonstrations using Deep Learning. In Joint IEEE International Conference on Development and Learning (ICDL) and Epigenetic Robotics (EpiRob). IEEE.

Fukui, R., Niho, T., Nakao, M. and Uetake, M., 2015. Imitation-based control of automated ore excavator to utilize human operator knowledge of bedrock condition estimation and excavating motion selection. In 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages. 5910-5916. IEEE.

Kim, J. S., Ahn, G. T., Park. S. Y., 2018. A Block-based Path Recognition of Slag Removal Using Convolutional Neural Network. In 2018 International Conference on Pattern Recognition and Artifical Intelligence (ICPRAI).

Minoura, H., Hirakawa, T., Yamashita, T. and Fujiyoshi, H., 2018. Path Predictions using Object Attributes and Semantic Environment. In 14th International Conference on Computer Vision Theory and Applications (VISIGRAPP), pages. 19-26.

Felzenszwalb, P.F. and Huttenlocher, D.P., 2012. Distance transforms of sampled functions. Theory of computing, volume 8.1, pages.415-428.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages. 770-778.

Clevert, D.A., Unterthiner, T. and Hochreiter, S., 2015. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289.

Zeiler, M.D., 2012. ADADELTA: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.