Recognition of Slab Identification Numbers using a Fully Convolutional Network

Sang Jun LEE,1) Wookyong KWON,2) Gyogwon KOO,1) Hyeyeon CHOI1) and Sang Woo KIM1)*

1) Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, 37673 South Korea. 2) Department of Creative IT Engineering, Pohang University of Science and Technology (POSTECH), Pohang, 37673 South Korea.

(Received on November 26, 2017; accepted on December 26, 2017; J-STAGE Advance published date: February 14, 2018)

In the steel industry, slabs are manufactured with different amounts of alloying elements according to production purposes or final products. Because slabs have similar shapes, product identification is required to prevent inadequate production processes. In many steel mills, paint marking systems are widely used to inscribe slab identification numbers (SINs). As smart factory technology receives more attention in recent years, automatic recognition of SINs becomes more important for factory automation. The recognition of SINs is a challenging problem due to complex background of factory scenes and low quality of characters in SINs. To address this difficulties, this paper proposes a deep learning algorithm for recognizing SINs in factory scenes. Most existing recognition algorithms conduct text detection and classification using separate modules, and errors in each step are accumulated. The proposed algorithm employs a fully convolutional network (FCN) with deconvolution layers to integrate the recognition processes and improve the performance in processing time and accuracy. The main contribution of this work is on a novel type of ground-truth data (GTD) for the training of a FCN to recognize SINs in factory scenes. The relation between an input image and the corresponding GTD is directly trained in the manner of image-to-image training, and the FCN generates a prediction map that contains categorical information of individual pixels in an input image. Experiments were thoroughly conducted on industrial data collected from a steelworks to demonstrate the effectiveness of the proposed algorithm.

KEY WORDS: steel industry; factory automation; product identification; deep learning; convolutional neural network; scene text recognition.

1. Introduction

Steel manufacturing is a fundamental industry in the world, and slab is one of the major steel products. In manufacturing process, individual slabs have similar shapes with different amounts of alloying elements according to production purpose or final product. Therefore, identification of individual slabs is required to prevent an incorrect production process. A widely used method in the steel industry is to inscribe a slab identification number (SIN) by using a paint marking machine, and automatic recognition of SINs is a key technology for factory automation. The objective of this paper is to develop a computer vision algorithm for recognizing SINs in factory scenes.

Our automatic recognition system for SINs contains an image acquisition device that was installed at a slab yard in an actual steelworks. A factory scene collected at the slab yard contains unknown number of slabs with different sizes of SINs, and a SIN consists of 9 characters that are horizontally placed with similar distances. The first character is an alphabet that represents a production line, and subsequent characters are numerals that characterize a slab. The recognition of SINs is a challenging problem due to complex background of factory scenes and change of lighting condition during day and night operation. Furthermore, extremely high temperature of slabs is another factor that causes difficulties for recognition. Characters in SINs are deformed or blurred due to the high temperature of slabs, and the quality of characters becomes worse during manufacturing processes. Example factory scenes with enlarged regions of SINs are presented in Fig. 1, and it shows difficulties for recognizing SINs.

The recognition of SINs can be viewed as a text recognition problem. Two typical methodologies for text recognition are stepwise and integrated approaches.1) In stepwise approach, algorithms for text localization, character segmentation, and recognition are separately developed, and these algorithms are sequentially processed in a recognition pipeline. Algorithms based on a sliding window method,2) maximally stable extremal regions,3) and projection profile analysis4) were proposed for text detection or text line segmentation. For text recognition, machine learning techniques with carefully engineered feature representations
were used. On the other hand, integrated approach employs a unified framework for detection and recognition by sharing text information in overall process. After the success of a convolutional neural network (CNN) for image classification and object recognition, CNNs were used to simultaneously detect and recognize characters in unstructured scenes.

These days, various computer vision algorithms are widely used to assist steel manufacturing processes. To extract slab information, rule-based algorithms were proposed to acquire the information of slabs at a reheating furnace and slab yard. Horizontal projection profile analysis and edge detection algorithm were used as a pre-processing, and carefully engineered score functions were employed to filter out false candidates. However, these previous algorithms should be modified for different target products or change of system environments, and manual tuning is required for many heuristic parameters. Recently, data-driven algorithms were proposed to address the limitations of rule-based methods for the localization of SINs. A sliding window method was used to extract patch images, and the patch images were classified to acquire slab information by using a conventional CNN classifier. However, the conventional CNN architecture for classification requires a pre-processing such as a sliding window method to extract character candidates, and it is computationally inefficient.

To address the limitations of the previous algorithms, this paper proposes a novel recognition pipeline for SINs in factory scenes. The proposed algorithm was inspired by a semantic segmentation algorithm that employs a FCN with deconvolution layers. FCN is an architecture of a CNN without the use of fully-connected layers to generate a multi-dimensional output prediction map. After the success of FCNs in object detection and edge detection, it was applied to the scene text localization. In this paper, a novel shape of GTD is proposed to train a FCN for the simultaneous localization and recognition of text information. Although several researches have been conducted to use FCNs for localizing text information, it is the first trial that applies a FCN for the recognition. Because SINs satisfy the sufficient conditions for applying the proposed algorithm, SINs of various numbers and sizes in factory scenes were successfully recognized. Experiments were thoroughly conducted on industrial data collected from an actual steelworks to verify the effectiveness of the proposed algorithm. The remaining sections are organized as follows. Section 2 explains the proposed recognition algorithm, and Section 3 presents experimental results. Section 4 contains a conclusion.

2. Proposed Algorithm

2.1. Ground-truth Data

For scene text recognition, a meaningful segment for a character is regarded as a collection of connected pixels that have consistent color information. However, blurred characters have low contrast to background and unclear boundaries. Furthermore, generating GTD with the exact shapes of individual characters as shown in Fig. 2(a) is an expensive and exhausting task. On the other hand, a bounding-box that contains a character or word has been considered as a meaningful region for solving the problem of scene text recognition. In this approach, the information of a text is recorded as a 4-tuple of (y-position, x-position, height, width) with corresponding character or word information as presented in Fig. 2(b). Based on these conventions, we propose a novel shape of GTD to train a FCN for the simultaneous localiza-

**Fig. 1.** Example factory scenes with enlarged SIN regions. (a) Various sizes of slabs and low contrast of characters due to shadow. (b) A low-quality SIN that has deformed or removed parts of characters. (c) Reddish color noise due to the high temperature of slabs. (Online version in color.)

**Fig. 2.** Meaningful information for scene text recognition. (a) Sample factory scene. (b) Collections of connected pixels with consistent color information. (c) Bounding-boxes for texts. (Online version in color.)
tion and recognition of text information.

The shape of the proposed GTD is an image that has identical height and width to the corresponding scene. A GTD for a sample image \( X_n = \{ x_{ij}^{(n)} : i=1, \cdots, H, j=1, \cdots, W \} \) in the training set \( T = \{ (X_n, Y_n) : n=1, \cdots, N \} \) is denoted by \( Y_n = \{ y_{ij}^{(n)} : i=1, \cdots, H, j=1, \cdots, W \} \), where \( H \) and \( W \) are the height and width of \( X_n \). \( N \) is the number of training samples, \( y_{ij}^{(n)} \in \{ 0, \cdots, K \} \) indicates the category of \( x_{ij}^{(n)} \), and \( K \) is the number of character classes. If \( X_n \) is a 24-bit color image, \( x_{ij}^{(n)} \) is a three dimensional vector such that \( x_{ij}^{(n)} = (r, g, b) \), and if \( X_n \) is a 8-bit gray image, \( x_{ij}^{(n)} \) is a pixel located in a bounding-box of a character \( c_k \in C = \{ c_k : k=1, \cdots, K \} \), where \( c_k \) is the \( k \)-th class of characters and \( C \) is the set of character classes. If \( x_{ij}^{(n)} \) does not belong to any bounding-box, then such \( y_{ij}^{(n)} \) is regarded as background, and the value of 0 is assigned to \( y_{ij}^{(n)} \). Since each image sample is processed independently, the superscripted \( n \) is omitted in the subsequent description for notational simplicity.

In our industry application, the first character of SINs is an alphabet, and it represents a production line. Because the dataset was collected at a production line which was represented by ‘B’, \( K \) was set to 11 for considering a kind of an alphabet, and it represents a production line. Because the shape of the proposed GTD is an image that has identical height and width to the corresponding scene.

A GTD image was horizontally partitioned into 10 sub-regions, and the number of positive elements in the \( l \)-th row of \( O \) is denoted by \( P_{H,l} \). \( P_{H,l} = \sum_{i=1}^{W} o_{ij} \). The vertical positions of SIN candidates can be obtained by \( I_v = \{ i : P_{H,i} > \eta \} \), where \( \eta \) is a heuristic threshold value. Let \( I_v^{(l)} \) be the \( l \)-th partition with connected elements of \( I_v \), then \( I_v^{(l)} \) represents the set of vertical indexes for the \( l \)-th SIN candidate. The sub-region that corresponds to the vertical indexes in \( I_v^{(l)} \) is extracted from the output prediction map \( O \).

The second step for the recognition is finding horizontal positions of characters in a sub-region of \( O \). Let \( O_{\text{sub}}^{(l)} \) be the
l-th sub-region of O such that \( O_{\text{sub}}^{(l)} = \{ o_{ij} \mid i \in I_{\text{sub}}^{(l)}, j=1, \ldots, W \} \). The number of positive elements in the j-th column of \( O_{\text{sub}}^{(l)} \) is denoted by \( P_{Vj}^{(l)} \), where \( P_{Vj}^{(l)} = \sum_{i \in I_{\text{sub}}^{(l)}} I_{o_{ij}} \). From \( P_{Vj}^{(l)} \), start and end point of the l-th SIN candidate in the horizontal direction is obtained by \( p_1 = \min I_{H}^{(l)} \) and \( p_2 = \max I_{H}^{(l)} \), where \( I_{H}^{(l)} = \{ j \mid P_{Vj}^{(l)}>0 \} \). The width of a character can be estimated by \( w = (p_2 - p_1)/10 \) by using the assumption that a SIN contains 9 equally-spaced characters and a blank between 6-th and 7-th characters.

The string information of the l-th SIN can be obtained by transcribing the elements in \( S^{(l)} \) such that \( S^{(l)} = \{ s_{ml} \mid 1 \leq m \leq 10, m \neq 7, m \in \mathbb{Z} \} \). \( m_{l} = \arg \max_{m} \sum_{i} I_{o_{ij}} \), and \( I_{H}^{(l)} = \{ p_1 + (m - 1) \cdot w, \ldots, p_1 + m \cdot w \} \). Figure 6 depicts the post-processing for transcribing SINs from an output prediction map of the FCN.

3. Experimental Results

3.1. Datasets

Factory scenes were recorded at a slab yard in an actual steelworks during 55 working days, and 4,501 scenes were collected as 24-bit color images with the size of 600 (height) × 960 (width). For the training and evaluation of the proposed algorithm, collected data were divided into training, validation, and test sets. The training data consisted of 1,850 scenes that were collected during 30 working days. Each factory scene contained one, two, or three slabs, and the training set contained 3,749 slabs. The validation set was collected during 6 days, and it contained 543 scenes with 1,102 slabs. The test set was collected during 19 days, and it contained 2,108 scenes with 4,275 slabs.

3.2. Experimental Environment and Evaluation Measures

Hardware environment for experiments included Intel core i7–6850 K CPU (3.6 GHz), 32 GB DDR3 RAM, and an NVIDIA TITAN X. Python and Tensorflow were used to implement the proposed algorithm. A SIN is a string of characters that characterize a slab, and correct recognition of entire characters in a SIN is required to identify a slab. Therefore, the recognition accuracy was measured based on the number of correctly recognized SINs. A correct recognition of a SIN means entire characters in the SIN were correctly recognized. Following three statistical measures were utilized to evaluate the proposed algorithm: sensitivity, precision, and f1-score. Sensitivity is the ratio of the number of correct predictions to the total number of predictions. Precision is the ratio of the number of correct predictions to the total number of predictions. F1-score is the harmonic mean of sensitivity and precision.

3.3. Training and Validation

The FCN was trained for 50 epochs using the training data, and 8.71 h was taken for the training process. Adam optimization was used with the initial learning rate of \( 10^{-4} \), and the learning rate was decreased by 5% for each epoch. Figure 7 presents learning rate and training loss during 50 epochs. During the training process, recognition accuracy for the validation set was evaluated to find an optimal training epoch. The heuristic parameter (\( \eta \)) in the post-processing for an output prediction map was set to 20, and this parameter is analyzed in Section 3.7. Figure 8 presents f1-scores for the training and validation sets during the training of 50 epochs. The f1-score for the validation set was maximized.
at the training of 18 epochs, and further training did not significantly improve the validation accuracy. At the training of 18 epoch, f1-score for the training set was 99.8242%, and f1-score for the validation set was 98.8203%.

3.4. Performance Comparison with Previous Algorithms

The main FCN model with the training of the optimal epoch was used to evaluate the proposed algorithm. For the test set, the main FCN model recorded the sensitivity of 99.1345% and precision of 99.1577% for the end-to-end recognition of SINs. Figure 9 presents the test accuracy for each epoch. The highest sensitivity and precision were achieved by 99.2749% and 99.2745% at the training of 15 epochs which was close to the optimal epoch obtained by investigating the validation accuracy. With the training of excessively many epochs, the recognition accuracy for the test set was slightly decreased. The averaged processing time for recognizing SINs in a factory scene was 0.0825 s. For a test image, 0.0801 s was taken in the inference procedure for the FCN, and 0.0082 s was taken for transcribing SINs from an output prediction map.

Although the end-to-end recognition includes localization task, the proposed algorithm achieved better results for the end-to-end recognition of SINs compared to the previous localization algorithms. Furthermore, compared to the recent algorithm, about 47% of error in the sensitivity was decreased from 1.64% to 0.8655% for the end-to-end recognition of SINs. Table 1 compares the performance of the proposed algorithm with the previous work for the identical data sets.

Major limitations of the previous algorithm are related to a fixed size of receptive field. Because a CNN was trained for classifying patch images of a fixed size, multi-scale analysis was required for recognizing SINs of various sizes. Furthermore, redundant computations for overlapped regions of adjacent sliding windows caused high computational complexity. To address these limitations, the proposed algorithm uses information from various receptive fields by combining mid-level feature maps. The use of a deep architecture with deconvolution layers is another reason for the improvement of recognition accuracy. Furthermore, by using a FCN architecture, the processing speed was remarkably improved by more than 150 times compared to the previous algorithm for the identical task.

3.5. Performance Comparison with Other FCN Models

Three other FCN models were designed and analyzed to develop the main FCN model presented in Section 2.3. These auxiliary FCN models are named model A, model B, and model C. Model A and model B are shallower, and model C is deeper than the main FCN model. The architectures of these FCN models are presented in Fig. 10.

These models are trained and evaluated in the same way as the main FCN model. Table 2 summarizes com-

![Fig. 9. Sensitivity and precision of the proposed algorithm for the test data.](image)

![Fig. 10. Three auxiliary FCN models.](image)

Table 1. Performance comparison with previous methods.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Choi et al.(^{17})</th>
<th>Lee &amp; Kim(^{18})</th>
<th>Lee et al.(^{19})</th>
<th>Proposed method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Rule-based method</td>
<td>CNN</td>
<td>CNN</td>
<td>FCN</td>
</tr>
<tr>
<td>Task</td>
<td>Localization</td>
<td>Localization</td>
<td>Recognition</td>
<td>Recognition</td>
</tr>
<tr>
<td>Sensitivity (%)</td>
<td>95.41</td>
<td>98.56</td>
<td>98.36</td>
<td>99.1345</td>
</tr>
<tr>
<td>Precision (%)</td>
<td>–</td>
<td>98.70</td>
<td>99.13</td>
<td>99.1577</td>
</tr>
<tr>
<td>Processing time (s)</td>
<td>–</td>
<td>1.19</td>
<td>13.91</td>
<td>0.0825</td>
</tr>
</tbody>
</table>
plexity, processing time, and recognition accuracy of the FCN models. The number of weight layers and parameters represents the complexity of FCN models. Training time was measured for the training of 50 epochs, and test time indicates the processing time for recognizing SINs in a test image. For each FCN model, an optimal training epoch was investigated by using the validation set, and the FCN with the training of each optimal epoch was used to evaluate recognition accuracy. F1-scores of model A, B, C for the validation set were maximized at the training of 17, 14, and 12 epochs. For the test set, sensitivity (precision) of model A, B, C, and the main FCN model was 98.3626% (97.9045%), 99.0877% (98.9488%), 98.6433% (98.6895%), and 99.1345% (99.1577%), respectively. Figure 11 presents F1-scores of the four FCN models for the test set at each epoch. Although model C contains more layers and parameters than the main FCN model, recognition accuracy was not significantly improved.

3.6. Effectiveness of the Transfer Learning

In the training of the FCN model, transfer learning of VGG19 was used to initialize the weights in the convolution layers. This section analyzes the effectiveness of the transfer learning. Figure 12 presents F1-score for the test set using the main FCN model which was trained with or without the use of pre-trained parameters. Without the use of pre-trained parameter, the F1-score for the validation set was maximized at the training of 34 epoch, and corresponding F1-score for the test set was 98.4799%. By using the pre-trained parameters of VGG19, the optimal training epoch was 18, and corresponding F1-score was 99.1461%. The averaged F1-score during the training of 50 epochs was 93.8662% (with transfer learning) and 88.3452% (without transfer learning). These results demonstrate that the use of the transfer learning reduces the training time and increases recognition accuracy, although the pre-trained parameters were trained on other domain.

3.7. Parameter Analysis

The proposed algorithm contains a heuristic threshold value (η) in the post-processing for finding vertical positions of SINs from an output prediction map. This section analyzes the effect of the heuristic parameter for the recognition accuracy. The major effect of the parameter is related to the number of predicted SINs. If the threshold value is small, many SIN candidates are generated, and it decreases precision rate. Figure 13 presents sensitivity and precision for the test set with various threshold values. Although the value of zero for the parameter caused slightly low precision, almost same sensitivity and precision were recorded for small and large positive values. This little variation on the sensitivity and precision demonstrates that the proposed algorithm is insensitive to the parameter η.

3.8. Result Images

The output of the proposed algorithm is transcribed SINs in an input factory scene. Figure 14 presents input image
samples and corresponding output prediction maps of the main FCN model. A correct or incorrect recognition result was transcribed by yellow or red color on each output prediction map. Figure 15 presents correct recognition results that were incorrectly recognized by the previous algorithms.\cite{18,19} Figure 16 presents result images that contain incorrectly recognized SINs. These result images show the effectiveness of the proposed algorithm for recognizing SINs in factory scenes.

5. Conclusion

This paper proposes a simple and effective recognition algorithm for a confined scope of texts that contain a known number of equally-spaced characters. An image-typed GTD is proposed for the scene text recognition; a semantic region of a character is considered as a rectangle region that contains the character, and the categorical information of the character is labeled inside the rectangle region. FCN with deconvolution layers was used to conduct image-to-image training of an input scene and its corresponding GTD. Feature maps were upsampled by deconvolution layers to generate an output prediction map with the same size as an input image. By combining mid-level feature maps, the information from various receptive fields was jointly used to recognize texts of various sizes. FCN models with various complexities were designed and analyzed to develop an appropriate FCN architecture for a specific application.

The proposed algorithm was applied to the recognition of SINs in factory scenes. A SIN is constructed with a

![Image of Figure 14](image14.png)

**Fig. 14.** Result images. (a) presents a SIN with blurred characters, and (b) contains a shadow on the bottom slab. (c) and (d) present factory scenes with slabs of various sizes, and (e) and (f) show reddish color noise due to the high temperature of slabs. (g) and (h) present diffused characters in SINs. (Online version in color.)

![Image of Figure 15](image15.png)

**Fig. 15.** Correct recognition results that were failed by the previous algorithms. The previous localization algorithm\cite{18} was failed for (a) and (b), and the end-to-end recognition algorithm\cite{19} was failed for (c) and (d). (Online version in color.)
fixed number of characters that are horizontally placed with similar distances. Therefore, this real industry problem satisfies the sufficient conditions for applying the proposed algorithm. The simple recognition pipeline and improved accuracy with significantly reduced processing time are main advantages of the proposed algorithm. Experiments were thoroughly conducted with actual factory scenes, and the performance was compared with previous algorithms that were carefully engineered for the identical dataset. The proposed algorithm is expected to be used for other industry applications such as license plate recognition and product serial number recognition.

REFERENCES

11) T. Wang, D. J. Wu, A. Coates and A. Y. Ng: 21st Int. Conf. on Pattern Recognition (ICPR 2012), ICPR 2012 Organizing Committee, Tsukuba, (2012), 3304.