Hindawi
Complexity
Volume 2021, Article ID 6691117, 10 pages
https://doi.org/10.1155/2021/6691117

Research Article

Image-Based Iron Slag Segmentation via Graph Convolutional Networks

Wang Long[1] **Zheng Junfeng**[2] **Yu Hong**[2] **Ding Meng**[3] **and Li Jiangyun**[2,4]

[1] *State Key Laboratory of Advanced Special Steel & Shanghai Key Laboratory of Advanced Ferrometallurgy & School of Materials Science and Engineering, Shanghai University, Shanghai, China*
[2] *School of Automation & Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China*
[3] *Scoop Medical, Inc., Houston 77007, TX, USA*
[4] *Shunde Graduate School of University of Science and Technology Beijing, Foshan 528000, China*

Correspondence should be addressed to Li Jiangyun; leejy@ustb.edu.cn

Received 10 October 2020; Revised 6 December 2020; Accepted 6 January 2021; Published 2 February 2021

Academic Editor: Ning Cai

Copyright © 2021 Wang Long et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Slagging-off (i.e., slag removal) is an important preprocessing operation of steel-making to improve the purity of iron. Current manual-operated slag removal schemes are inefficient and labor-intensive. Automatic slagging-off is desirable but challenging as the reliable recognition of iron and slag is difficult. This work focuses on realizing an efficient and accurate recognition algorithm of iron and slag, which is conducive to realize automatic slagging-off operation. Motivated by the recent success of deep learning techniques in smart manufacturing, we introduce deep learning methods to this field for the first time. The monotonous gray value of industry images, poor image quality, and nonrigid feature of iron and slag challenge the existing fully convolutional networks (FCNs). To this end, we propose a novel spatial and feature graph convolutional network (SFGCN) module. SFGCN module can be easily inserted in FCNs to improve the reasoning ability of global contextual information, which is helpful to enhance the segmentation accuracy of small objects and isolated areas. To verify the validity of the SFGCN module, we create an industrial dataset and conduct extensive experiments. Finally, the results show that our SFGCN module brings a consistent performance boost for a wide range of FCNs. Moreover, by adopting a lightweight network as backbone, our method achieves real-time iron and slag segmentation. In the future work, we will dedicate our efforts to the weakly supervised learning for quick annotation of big data stream to improve the generalization ability of current models.

1. Introduction

Slagging-off is an essential operation in steel-making. It is used to remove high sulfur slag from molten iron to improve the purity of iron. The process of slagging-off is shown in Figure 1(a) and the actual image obtained by video capture is shown in Figure 1(b). In this process, molten iron is inevitably brought out, and the loss of molten iron is directly proportional to the clean rate of slagging-off. Meanwhile, slagging-off operation will be accompanied by the decrease of molten iron temperature. Therefore, accuracy and efficiency are two key factors of slagging-off operation, which are directly related to production energy consumption. At present, manual operation of machinery for slag removal is a commonly employed scheme in industrial applications. However, affected by the long-term strong light and dense smoke condition, it can easily lead to misidentification and misoperation. Besides, manual operation is inefficient. With the introduction of Industry 4.0 paradigm, the trend is moving towards to intelligent production line, where automatic slagging-off will benefit the modern smart manufacturing greatly.

Recognition of iron and slag is the premise of automatic slagging-off operation. We formulate this problem as a semantic segmentation task, which is a fundamental problem of computer vision and aims to assign categories for

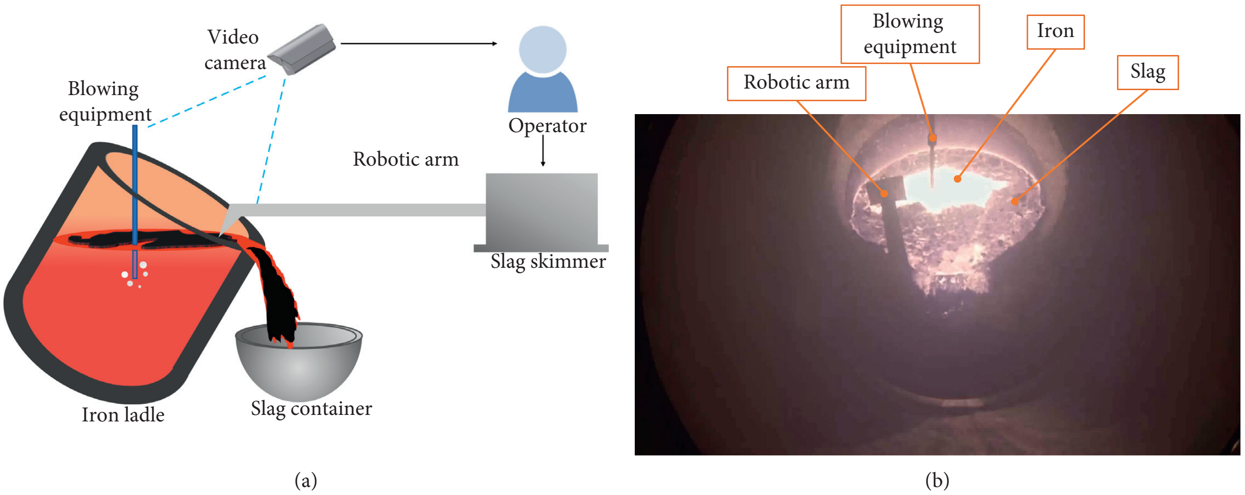

Figure 1: (a) The process of slagging-off. The operator obtains real-time images of the iron ladle through the video camera and operates the slag skimmer to move the slag to the slag container. When the slag is less or dispersed, the blowing equipment will blow nitrogen to polymerize the slag. During this process, it is necessary to avoid the collision between the blowing equipment, the inner wall of the iron ladle, and the robotic arm. (b) The image of actual slagging-off operation working condition from video capture.

each pixel in an image. Many classical machine vision methods have been proposed for image segmentation. However, the monotonous gray value of industry images and poor image quality caused by strong light and dense smoke condition challenge the performance of traditional computer vision algorithms. As far as the task of iron and slag segmentation is concerned, results of some traditional algorithms including K-means, Markov random field, and mean shift are shown in Figure 2, which are obviously cannot meet the requirement of industrial application. Currently, the state-of-the-art methods for segmentation mainly based on fully convolutional networks (FCNs) are used [1]. However, only modeling local correlation with convolutional operations, FCNs are not effective to reason relation between distant regions with arbitrary shape without stacking multiple convolution layers. To tackle this problem, many algorithms have been proposed to expand the receptive field of FCNs to capture long-range contextual information in the scene. Dilated convolution has been implemented to capture large objects, thus introducing another problem that small objects may be ignored. Another research direction is fusing multiscale features [2], which is inefficient. Recently, self-attention mechanism-based methods [3] make use of affinity matrix to model the relation between each spatial position and its neighborhoods. However, the memory and computational requirements of large affinity matrix prevent the application of these methods for high-resolution image segmentation application, such as the iron and slag segmentation with a resolution of 1920 × 1080.

The monotonous gray value of industry images, poor image quality, and irregular and scattered shape of slag also challenge the existing FCNs. Graph convolution is an efficient and effective operation to model global contextual information over regions in a single layer, which has been widely employed in recent scene understanding works [4, 5]. Motivated by these works, we propose an effective and efficient spatial and feature graph convolutional network (SFGCN) module based on graph convolution. Different from previous works, our SFGCN module makes use of latent interaction space to efficiently perform global reasoning function. Our SFGCN module consists of two parallel branches to project feature maps to latent spatial space and feature space, respectively. Then, graph convolutions are employed to perform relation reasoning. After graph reasoning, the updated information is reprojected back into the original coordinate space for further information extraction. Extensive experiments prove that our SFGCN module can consistently improve the performance of current mainstream convolutional neural network backbones for iron and slag segmentation.

Our contributions can be summarized as follows:

(1) We formulate the slagging-off problem as an image-based semantic segmentation task and explore deep learning methods to tackle the automatic iron and slag recognition task for the first time.

(2) Considering the limitation of convolution operations for modeling local correlation, we propose a SFGCN module to effectively reason global information interaction via weighted spatial graph convolution and feature graph convolution branches. The proposed network is termed as SFGCNet.

(3) We establish an industrial slagging-off dataset and conduct extensive experiments, and the results show that our SFGCN module brings consistent performance improvement for a wide range of network backbones for iron and slag segmentation. Moreover, taking a lightweight network as backbone, our method is able to achieve real-time segmentation of iron and slag.

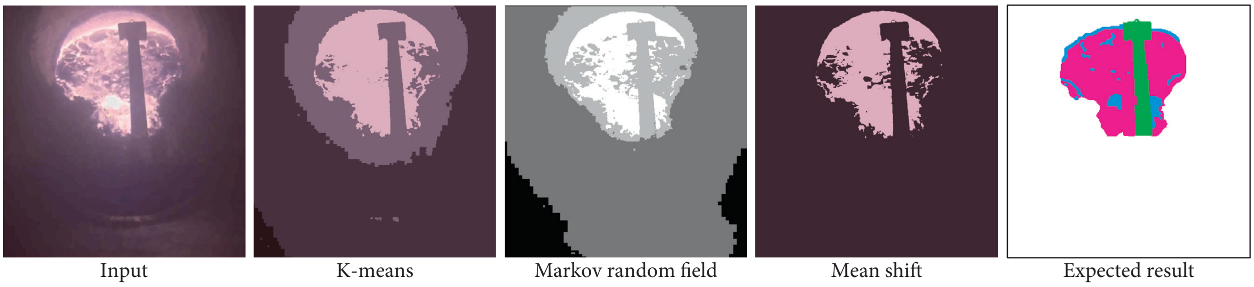

Figure 2: The segmentation results of traditional methods including K-means, Markov random field, and mean shift. The last column is the segmentation we expect. In the expected result, white, green, blue, and pink represent background, robotic arm, iron, and slag, respectively.

2. Related Work

Fully convolutional networks (FCNs) have made great progress in semantic segmentation [1,6]. There are many variants to improve the performance of segmentation; we briefly review several main research directions in scene understanding domain, including network architecture implementation, global context reasoning, and graph-based reasoning.

2.1. Network Architecture Implementation. Atrous Spatial Pyramid Pooling (ASPP) has been proposed and employed in Deeplabv2, v3 [7] to integrate multiscale contextual information, which contains multiple parallel dilated convolutions with different dilated rates. A variety of encoder-decoder structures have been implemented to obtain effective usage of midlevel and high-level extracted features [8, 9]. PSPNet [2] builds a novel pyramid pooling module to get multiscale contextual prior knowledge. DenseASPP [10] embeds multiscale features to expand the receptive field of convolution layers for segmentation task. All these methods effectively stack multiple convolution layers to collect multiscale information.

2.2. Global Context Reasoning. Many methods have been proposed to overcome the limitation that convolution layers are difficult to capture global context, such as self-attention mechanism and nonlocal networks. Self-attention mechanism is firstly proposed in [11] to model long-range dependencies for machine translation task and has been widely applied in many tasks in recent years [12]. PSANet [13] captures pixel-wise relation by applying attention module in spatial dimension. EncNet [14] and DFN [15] apply attention module along the channel dimension of the feature map to account for global context. DANet [16] uses attention module in both spatial and channel dimensions. Nonlocal networks [3, 17] aim to deliver long-range information from one position to another.

2.3. Graph-Based Reasoning. Graph-based reasoning provides an efficient idea of global context reasoning. Random walk and conditional random field (CRF) networks have been proposed based on graph for efficient image segmentation and classification. Recently, graph convolutional networks (GCNs) have been proposed for semisupervised image classification. Wang et al. [18] apply GCN to capture global contextual relation in video recognition task. Chen et al. [4] explore GCN to reason global relation in semantic segmentation task. Yan et al. introduce GCN to describe skeleton connections for action recognition [19, 20]. Following these methods, we propose a novel dual GCN module consists of spatial graph convolution and feature graph convolution to model global contextual information for iron slag segmentation. Our SFGCN module makes use of latent spatial and feature spaces to efficiently realize global relation reasoning, which alleviates the memory and computation burden of global context reasoning while improving the performance of segmentation.

3. Methods

In this section, we first review the graph convolution and then introduce the implementation of our SFGCN module. Finally, we detail the network architecture for slag segmentation.

3.1. Graph Convolution. Graph convolution is an efficient operation to reason global context information, which overcomes the limitation that convolution operation can only model local context information. Graph convolution defined in graph G with nodes N and edges E can effectively achieve global information interaction in a single operation. The specific operation can be defined as follows:

$$O = \sigma(AXW). \tag{1}$$

The specific implementation steps are shown as the following pseudocode, including (1) project the feature map from coordinate space to graph space, we employ the conventional convolution operations to project the feature map to graph space after the feature extraction operation, and the process is shown in Figure 3; (2) build adjacency matrix to describe intrabody connections of nodes within the graph; (3) update the weight matrix; and (4) reproject the graph to coordinate space. The feature map extracted by backbone networks contains spatial and channel dimensions. Assuming that spatial dimension abstracts the objects in the scene and channel dimension encapsulates the detailed object features, that means the graph established in spatial space is able to describe the relevance between objects

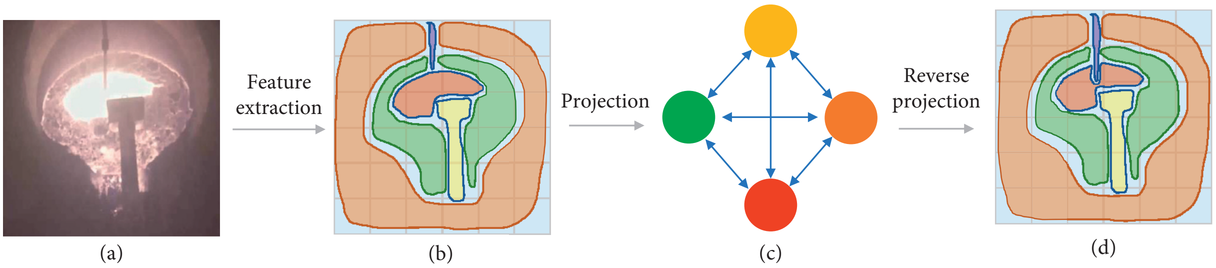

Figure 3: The illustration of graph convolution operation. (a) The input image. (b) The feature map obtained by backbones. (c) Projecting the feature map to graph space and reasoning the relation between all nodes. (d) Reprojecting the learned interaction to feature map for performance improvement. Here, we use nodes with different colors to represent different object regions.

in the scene and the graph established in feature space is able to express the relevance between object parts. Therefore, we conducted graph convolution on spatial graph and feature graph, respectively. The spatial branch is used to grasp thecinternal integrity of objects and the relationship between objects. The feature branch is used to characterize the details of objects and the relationship between features (Algorithm 1) [21].

3.2. Graph Convolution in Spatial Space

3.2.1. Spatial Space Projection. As shown in Figure 4, before conducting graph convolution operation, we first project the input feature map to latent spatial space to get the graph. In practice, spatial downsampling operation T_s is employed to transform the input feature $X \in R^{aH\times W\times C}$ to graph $G_s \in R^{(H\times W/d^2)\times C}$ in the latent spatial space S_s, where d represents the downsampling rate. We achieve T_s based on stacked depth-wise convolution operations in each layer with a stride of 2 and kernel size of 3×3. Then, G_s is obtained via

$$G_s = T_s(X). \tag{2}$$

3.2.2. Spatial Graph Convolution. After projecting the input feature X to graph G_s, the graph consists of $(H/d) \times (W/d)$ nodes. Each node of the graph integrates the information of a cluster of pixels in the feature map. To measure the correlation between nodes, we form an adjacency matrix $A_s \in R^{(HW/d^2)\times(HW/d^2)}$. The spatial graph convolution is implemented according to the following formulation to achieve global relation reasoning:

$$O_s = f\left(\delta_s(G_s) \cdot \psi_s(G_s)^T\right) \cdot G_s W_s, \tag{3}$$

where $f(\delta_s(G_s) \cdot \psi_s(G_s)^T)$ gives the adjacency matrix A_s and $f(\cdot)$ represents the softmax activation function, in which $\cdot$ is the dot-production operation. W_s is the weight matrix for updating information.

3.2.3. Reprojection. After relational reasoning, we reproject O_s back to the original coordinate space ($R^{H\times W\times C}$) for compatibility with later operations. Different from the downsampling operation T_s in graph projection, we directly employ nearest neighbour interpolation to upsample O_s to the original input size. Finally, the output feature map of spatial graph convolution branch is obtained according to $\widetilde{O}_s = \xi_s(\text{interp}(O_s))$.

3.3. Graph Convolution in Feature Space. Spatial graph convolution models the spatial correlation of pixel clusters in a scene, which enables the network to make correlation prediction based on all objects in the whole scene. Next, we consider projecting input feature map to feature space and reasoning correlation along the channel dimension. Assuming that the latter layers of the FCN are responsive to the object parts and high-level semantic features, conducting GCN in feature space can model the correlation of abstract features such as object parts. We first adopt a channel downsampling operation $\theta(\cdot)$ to reduce the channels of input feature from $X \in R^{H\times W\times C}$ to $H_f \in R^{H\times W\times C_1}$ and employ a linear combination function $\varphi(\cdot)$ to aggregate information along the channel dimension. Finally, we obtain the formulation of input feature $X \in R^{H\times W\times C}$ to feature space graph $G_f \in R^{C_1\times C_2}$:

$$G_f = \theta(X)^T \cdot \varphi(X) = H_f^T \cdot \varphi(X), \tag{4}$$

where C_1 represents nodes and C_2 denotes the states of each node. After feature space projection, the feature graph convolution and reprojection are conducted according to the following equations:

$$\begin{aligned} O_f &= \left(I + A_f\right) G_f W_f, \\ \widetilde{O}_f &= \xi_f\left(H_f \cdot O_f\right). \end{aligned} \tag{5}$$

Considering the low dimension of feature graph, we employ two 1D convolution layers as adjacent matrix A_f and trainable edge weights W_f. To alleviate the optimization difficulty, the adjacent matrix A_f is updated with a residual structure and reconstructed as $(I + A_f)$. Both A_f and W_f are randomly initialized and optimized with gradient descent during the training process.

3.4. SFGCNet. Finally, the output of SFGCN is computed as $\widetilde{X} = \omega X + \omega_s \widetilde{O}_s + \omega_f \widetilde{O}_f$, where "+" denotes point-wise summation and ω is the learnable weight coefficient. Now we can easily embed our SFGCN module into the existing network backbone (e.g., FCN and ResNet).

```
Input: Tensor extracted by convolutional network
Output: Tensor after graph convolution operation
function SFGCN(Tensor)
    Project coordinate input to graph space $X \leftarrow \mathrm{Projecting}(\mathrm{Tensor})$
    $\delta \leftarrow \mathrm{Conv}(X)$
    $\phi \leftarrow \mathrm{Conv}(X)$
    Build adjacency matrix $A \leftarrow \mathrm{Soft\,max}(\delta^T * \phi)$
    Update weight matrix $\mathrm{AXW} \leftarrow \mathrm{Conv}(A, X)$
    $O \leftarrow \mathrm{Activation}(\mathrm{AXW})$
    Reproject graph $O$ to coordinate space result $\leftarrow \mathrm{Reprojecting}(\mathrm{O})$
    return result
end function
```

Algorithm 1: GCN 1: realization of graph convolution.

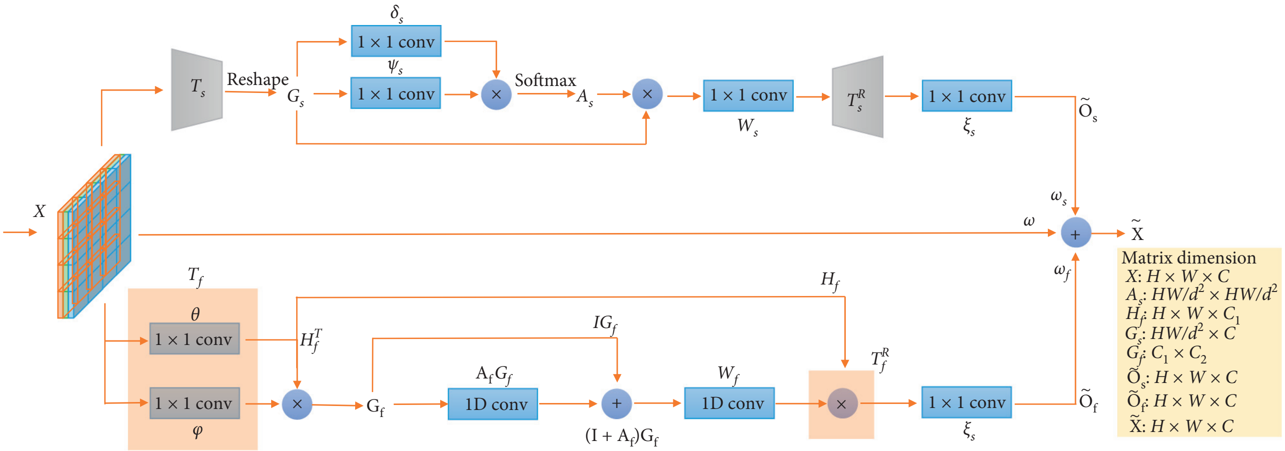

Figure 4: The design details of the SFGCN module. Our method contains two branches of graph convolution operation to model global contextual information along spatial and channel dimensions of feature map X.

3.4.1. Implementation of SFGCNet. As shown in Figure 5, we embed SFGCN module in the last stage of fully convolutional networks (FCNs) to achieve the segmentation of iron and slag. In order to verify the effectiveness of SFGCN module, we construct SFGCNet by adopting FCN [1], BiSeNet [22], ICNet [23], and ResNet-50 [24] as the network backbones, respectively. BiSeNet and ICNet are two lightweight networks to achieve real-time semantic segmentation.

4. Experiments

4.1. Dataset and Evaluation Metrics. As there has no public slagging-off dataset, we collect 7 videos from different industrial cameras. Due to the time-consuming and laborious segmentation labeling, we only select 24 clips from all 7 videos randomly. Each of the clips contains 64 frames. All of these clips are segmented with Photoshop software manually, by three raters, following the same annotation protocol, and their annotations are approved by experienced workers, and then, we split these images into training set and test set with a ratio of 3: 1. The annotation sample is presented in Figure 6. The training set is used to train models, and the test set is used to validate the performance of trained models.

The efficiency and accuracy of the model are mainly considered in industrial applications. The efficiency of the model can be evaluated by inference time, model parameters amount, and the total number of floating-point operations per second (FLOPs). To evaluate the accuracy of the model, we adopt the commonly used metrics in the segmentation task, including Mean Intersection over Union (MIoU) and pixel accuracy (PA). The two metrics are defined as follows:

$$\mathrm{MIoU} = \frac{1}{K+1}\sum_{i=0}^{K}\frac{P_{ii}}{\sum_{j=0}^{K}P_{ij} + \sum_{j=0}^{K}P_{ji} - P_{ii}},$$
$$\mathrm{PA} = \sum_{i=0}^{K}\sum_{j=0}^{K}\frac{P_{ii}}{P_{ij}}, \qquad (6)$$

where P_{ii} represents the pixel predicted correctly (i.e., the true category of the pixel is class i, and the prediction is class i too). P_{ij}, P_{ji} mean the pixel prediction is wrong (i.e., the true category of the pixel is class (i/j), and the prediction is class (j/i).

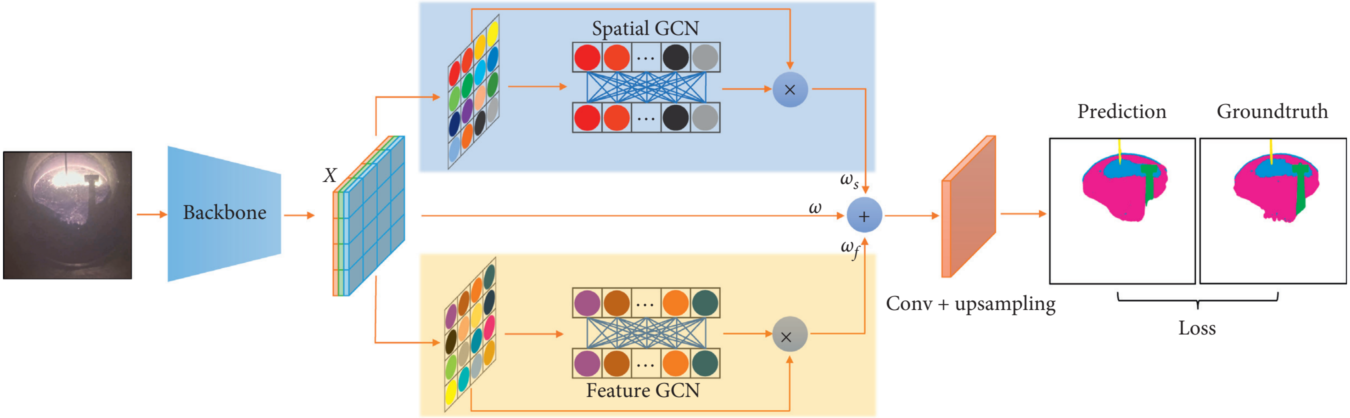

Figure 5: The architecture of the proposed network, i.e., SFGCNet. SFGCN module is inserted in the last stage of fully convolutional networks. The weights of SFGCNet are optimized by gradient descent algorithm and the cross entropy loss between prediction and groundtruth.

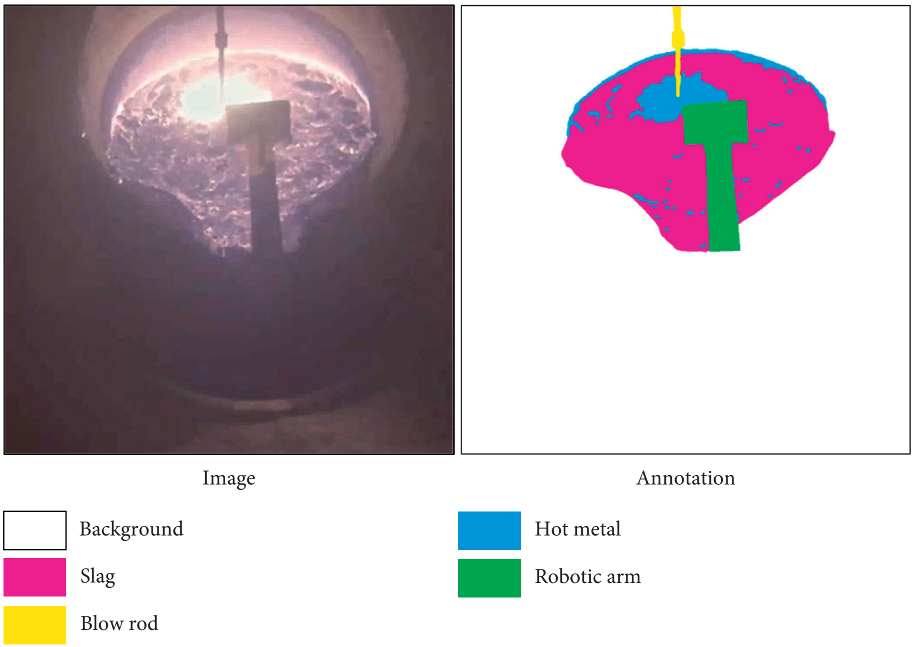

Figure 6: Image annotation for semantic segmentation. There are 5 categories in annotations, including white for background, pink for slag, yellow for blowing equipment, blue for iron, and green for robotic arm.

4.2. Preprocessing. The annotation of semantic segmentation is time-consuming and labor-intensive. Also, it is difficult to obtain a large number of labeled data in industrial applications. Thus, data augmentation is an effective method to expand the dataset, which is helpful for alleviating the overfitting problem and enhancing the robustness of the network. Considering that images acquired by the video camera contain a large number of background areas, which cannot benefit the accuracy, we firstly crop the raw image from 1920 × 1080 to 1024 × 1024 to reduce the proportion of background area. After that, we randomly apply the data augmentation methods with 50% probability, including the following:

(1) Random horizontal and vertical flips

(2) Random scaling between $[0.5, 2]$

(3) Random intensity shift between $[-0.1, 0.1]$

Table 1: The results of deep learning-based methods on the test set. The size of the input image is 1024 × 1024.

Models	Iron	Slag	Robotic arm	Blow pole	MIoU (%)	PA (%)	Inference time (ms)	Parameters (M)	FLOPs (G)
BiSeNet [22]	67.49	82.91	82.25	55.77	72.11	97.04	15.47	12.42	48.77
BiSeNet + SFGCN	64.55	82.98	80.66	71.16	**77.74**	97.26	18.28	13.4	60.35
ICNet [23]	60.74	69.54	67.17	72.92	67.59	94.61	44.62	28.29	147.68
ICNet + SFGCN	61.47	73.54	68.63	70.05	**68.42**	95.45	45.79	28.79	153.0
FCN [1]	68.22	83.08	83.62	64.63	74.89	97.15	66.67	18.64	321.78
FCN + SFGCN	68.24	83.76	84.92	71.23	**78.38**	97.31	67.46	21.86	324.52
ResNet-50 [24]	55.06	78.97	79.15	71.32	71.13	96.38	30.18	28.51	98.18
ResNet-50 + SFGCN	66.29	73.04	76.92	72.49	**75.00**	96.62	30.73	28.75	98.57

4.3. Experiments and Results

4.3.1. Experiment Setup. We implement our method with PyTorch. Cosine annealing learning rate policy is used with 30 warming-up epochs. The initial learning rate lr_0 is set to 0.001 and adjusted based on the following formulation:

$$\mathrm{lr} = \begin{cases} \mathrm{lr}_0 * \left(1 - \cos\left(\dfrac{\pi}{2} * \dfrac{e}{w}\right)\right), & e \leq w, \\ \\ \mathrm{lr}_0 * \dfrac{\cos\left(\pi^* (e - w)/t\right) + 1)}{2} & w < e < t. \end{cases} \tag{7}$$

Specifically, e represents the current training epoch, t is the total number of training epochs, and w denotes the warming-up epochs. We train our model with Adam optimizer and synchronized BN on four parallel Nvidia 2080Ti GPUs for 300 epochs. The batch size is set to 8 to guarantee the performance of batch normalization.

4.3.2. Experiment Results. We apply our SFGCN module to the last stage of typical backbones such as FCN, ResNet-50, BiSeNet, and ICNet to reason long-distance dependencies. Considering the distribution difference between industrial dataset and natural scene dataset, we train all the above backbones from scratch. As shown in Table 1, our SFGCN module widely improves the performance of different backbones. In terms of MIoU, SFGCN module brings 5.63%, 0.83%, 3.49%, and 3.87% improvements on BiSeNet, ICNet, FCN, and ResNet-50, respectively. Benefited from the global reasoning function of graph convolution, SFGCN module makes the isolated slag and molten iron region more easily to be identified. As shown in Figure 7, while the dispersed areas of slag and iron are easy to be segmented incorrectly, SFGCN alleviates the influence of neighbor regions on the classification of these regions. On the other hand, the introduction of SFGCN module only results in 2.81 ms, 1.17 ms, 0.79 ms, and 0.55 ms more inference time for BiSeNet, ICNet, FCN, and ResNet-50, respectively, as well as slight parameters and FLOPs increase, which demonstrates that our SFGCN module is efficient. Especially, taking lightweight BiSeNet as the backbone, our SFGCNet achieves real-time segmentation of iron and slag.

4.3.3. Ablation Studies and Discussion. Embedded location of SFGCN module: our SFGCN module can be flexibly embedded in any stage of the network backbone, and it is worth exploring where the embedding can achieve better results. Moreover, the embedding location will affect the accuracy and efficiency of the network at the same time. The feature map of shallow layers has high resolution, which directly increases the parameters and FLOPs of the SFGCN module. From the perspective of feature extraction, shallow layers cannot capture abundant semantic information due to the lacking of enough receptive fields, which will also limit the performance of SFGCN module. Experiments show that the SFGCN module achieves higher efficiency when it is embedded in the last stage of various backbones.

The effectiveness of each branch: to verify the effectiveness of SGCN branch and FGCN branch, we conduct experiments on BiSeNet and FCN with different settings in Table 2.

As shown in Table 2, both SGCN and FGCN boost the performance of BiSeNet and FCN. The introduction of SGCN and FGCN, respectively, yields 3.82% and 4.46% improvement in MIoU for baseline of BiSeNet. Meanwhile, SGCN and FGCN outperform the FCN baseline by 2.15% and 2.81%. After integrating SGCN and FGCN branches, our method achieves 5.63% and 3.49% performance boost for BiSeNet and FCN. Results show that SFGCN module brings benefits for the segmentation of iron and slag.

The effects of SGCN and FGCN branches are visualized in Figure 8. As shown in the third column, SGCN aggregates information of pixel cluster and delivers messages between nodes, thus guaranteeing the integrity of objects. However, spatial branch loses details of each node while aggregating node information. The FGCN branch focuses more on reasoning the details of objects to make up for the deficiency of the SGCN branch which focuses more on connection between objects. The refinement of segmentation is significantly improved as shown in the fourth column.

We compute the coefficients of SGCN and FGCN branches to objectively evaluate the contribution of these two branches. The shortcut connection weight ω is set to 1. ω_s and ω_f are initialized as 1 and learnable. The final coefficients of each branch of the SFGCN module in different backbones are shown in Table 3, and the results show that SGCN and FGCN branches do provide extra information for the segmentation. Moreover, the coefficients vary for different network backbones. Therefore, the learnable coefficients provide the flexibility of adjusting the contribution of each branch based on the information learned by the base network.

Effect of projection: as described in Section 3, we aggregate information along spatial and channel dimensions to

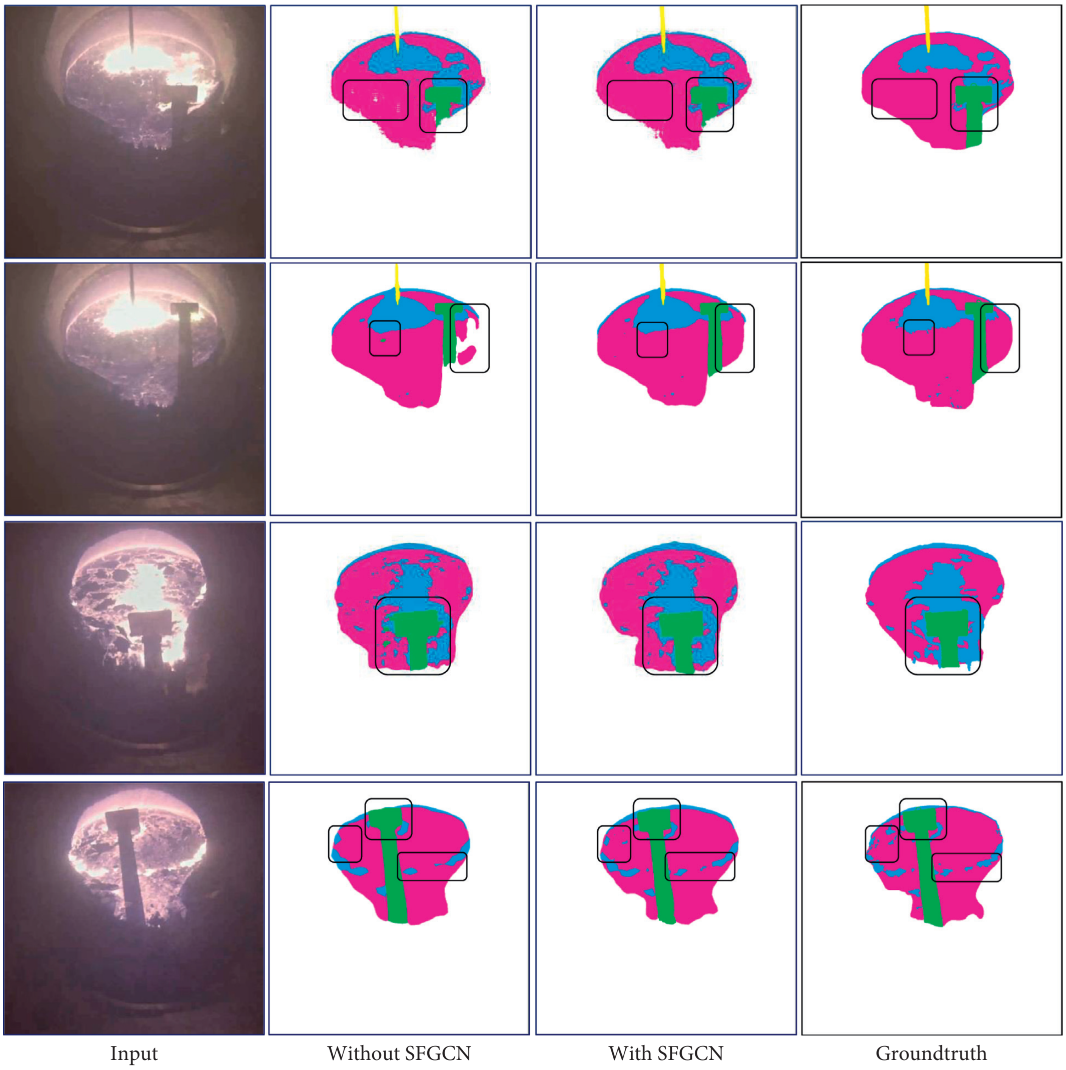

Figure 7: The visualization results of SFGCNet with different backbones. The results of FCN, BiSeNet, ICNet, and ResNet-50 are shown from the first row to the last row. More visual results are available at https://github.com/ustbzjf1/SFGCNet-for-hot-metal-slag-segmentation.

Table 2: The ablation study based on the network backbone of BiSeNet and FCN on the test set.

Backbone	SGCN	FGCN	MIoU
BiSeNet			72.11
BiSeNet	√		75.93
BiSeNet		√	76.57
BiSeNet	√	√	77.74
FCN			74.89
FCN	√		77.04
FCN		√	77.70
FCN	√	√	78.38

SGCN and FGCN represent spatial GCN and feature GCN, respectively.

project the input feature map to the graph space. The downsampling ratio of SGCN branch directly determines the degree of spatial information aggregation. Large ratio loses details while small ratio retains useless information. The number of nodes in the FGCN branch also affects the relation reasoning between the features of objects. Appropriate number of nodes is important for recovering the details of each object. After conducting extensive experiments on our dataset, we observe that SFGCN module brings more performance improvement when the size of G_s

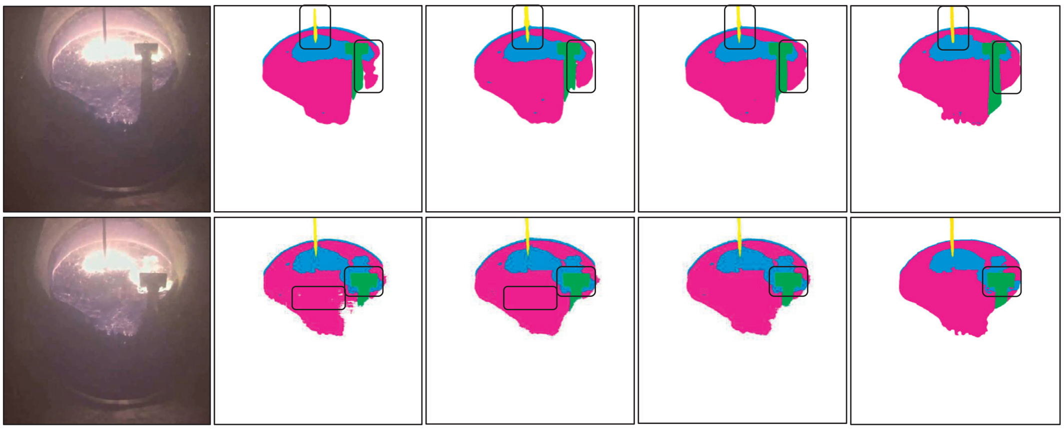

Figure 8: The visualization results of SGCN and FGCN on the test set.

Table 3: The adaptive coefficients of SGCN and FGCN branches.

SFGCNets	ω_s	ω_f	MIoU
BiSeNet + SFGCN	0.452	0.286	77.74
ICNet + SFGCN	0.143	0.712	68.42
FCN + SFGCN	0.320	0.438	78.38
ResNet-50 + SFGCN	0.897	0.526	75.00

is (1/64) of the input image size and the number of nodes for G_f is 32. We speculate that 64× downsampling to aggregate information is more suitable for the scale of objects and 32 nodes can better express the details of objects in the slagging-off scene.

5. Conclusion

In this work, we explore deep learning methods for iron and slag recognition. We formulate this problem as a semantic segmentation task and propose a SFGCN module to reason global contextual information according to the characteristic of the slagging-off task. Extensive experiments have verified that our method not only triumphs over traditional segmentation methods but also widely improves the performance of current mainstream deep learning models in the slagging-off task. Taking lightweight network as backbone, our SFGCNet can realize real-time and accurate recognition of iron and slag, which provides a significant reference for downstream automatic slagging-off operation.

Although our algorithm has achieved satisfactory results in view of accuracy and efficiency, we need to expand the dataset to improve the performance of the model in more scenarios. It is difficult to label industrial big data manually, in the future work, we will dedicate our efforts to the weakly supervised learning for quick annotation of big data stream to improve the generalization ability of current models.

Data Availability

The raw/processed data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by State Key Laboratory of Advanced Special Steel, Shanghai Key Laboratory of Advanced Ferrometallurgy and the Science and Technology Commission of Shanghai Municipality (No. 19DZ2270200), the Fundamental Research Funds for the China Central Universities of USTB (FR-DF-19-002), and Scientific and Technological Innovation Foundation of Shunde Graduate School, USTB (BK20BE014).

References

[1] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, Seattle, WA, USA, June 2015.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, Honolulu, HI, USA, July 2017.

[3] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, Salt Lake, UT, USA, October 2018.

[4] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, “Graph-based global reasoning networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 433–442, Long Beach, CL, USA, September 2019.

[5] X. Liang, Z. Hu, H. Zhang, L. Lin, and E. P. Xing, “Symbolic graph reasoning meets convolutions,” *Advances in Neural Information Processing Systems*, vol. 17, pp. 1853–1863, 2018.

[6] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: a deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[7] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 834–848, 2017.

[8] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, Honolulu, HI, USA, July 2017.

[9] K. Sun, B. Xiao, D. Liu, and J. Wang, “Deep high-resolution representation learning for human pose estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5693–5703, Long Beach, CL, USA, July 2019.

[10] M. Yang, K. Yu, C. Zhang, Z. Li, and K. Yang, “Denseaspp for semantic segmentation in street scenes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3684–3692, Salt Lake, UT, USA, October 2018.

[11] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 5998–6008, Barcelona, Spain, May 2017.

[12] T. Shen, T. Zhou, G. Long, J. Jiang, S. Pan, and C. Zhang, “Disan: directional self-attention network for rnn/cnn-free language understanding,” in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, February 2018.

[13] H. Zhao, Y. Zhang, S. Liu et al., “Psanet: point-wise spatial attention network for scene parsing,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 267–283, Venice, Italy, October 2018.

[14] H. Zhang, K. Dana, J. Shi et al., “Context encoding for semantic segmentation,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 32, pp. 7151–7160, 2018.

[15] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “Learning a discriminative feature network for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1857–1866, Salt Lake, UT, USA, June 2018.

[16] H. Nam, J. W. Ha, and J. Kim, “Dual attention networks for multimodal reasoning and matching,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 299–307, Honolulu, HI, USA, July 2017.

[17] Y. Yuan and J. O. Wang, “Object Context Network for Scene Parsing,” 2018, https://arxiv.org/abs/1809.00916.

[18] X. Wang and A. Gupta, “Videos as space-time region graphs,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 399–417, Glasgow, UK, August 2018.

[19] S. Yan, Y. Xiong, and D. Lin, “Spatial temporal graph convolutional networks for skeleton-based action recognition,” in *Proceedings of the Thirty-second AAAI conference on artificial intelligence*, New Orleans, LI, USA, February 2018.

[20] L. Shi, Y. Zhang, J. Cheng, and H. Lu, “Two-stream adaptive graph convolutional networks for skeleton-based action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12026–12035, Long Beach, CL, USA, September 2019.

[21] S. Zhang, H. Tong, J. Xu et al., “Graph convolutional networks: a comprehensive review,” *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.

[22] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, “Bisenet: bilateral segmentation network for real-time semantic segmentation,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 325–341, Munich, Germany, September 2018.

[23] H. Zhao, X. Qi, X. Shen, J. Shi, and J. Jia, “Icnet for real-time semantic segmentation on high-resolution images,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 405–420, Munich, Germany, September 2018.

[24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.