A Weakly-Supervised Semantic Segmentation Approach Based on a Closed-Form Matting Solution

Kai Yao, Alberto Ortiz, and Francisco Bonnin-Pascual,

***Abstract*—Weakly-supervised semantic segmentation is a challenging task owing to the complex content with non-uniform intensity distribution, diversity of targets, and lacking the sufficient and accurate ground truth. In order to overcome these challenges, we propose and assess a novel scribble-based weakly-supervised semantic segmentation solution by employing a closed-form matting loss function, namely closed-form matting (CFM) loss, which aims at propagating the semantic information from user annotations to unknown areas. Our solution only needs scribble annotations to train a segmentation network. Addressing the Multi-Category problem, a simple network, namely Multi-Label Classification (MLC) model, is proposed and integrated into the segmentation backbone. The performance of our approach is evaluated against datasets from two different industry-related inspection tasks: one task is designed to detect the marine creature attached to the hull of vessels through photos taken underwater; another task is to deal with the localization of image areas whose pixels correspond to scene surface points affected by a specific sort of defect. Besides, two datasets from PASCAL VOC and CityScape benchmarks are used to evaluate the performance of our solution for the multi-category problem. Experimental results exhibit that our approach can obtain reasonable segmentation performance only using scribble annotations.**

***Index Terms*—Object recognition, inspection, closed-form image matting, and weakly-supervised semantic segmentation**

I. Introduction

Image semantic segmentation is a fundamental problem in the computer vision area, which refers to a pixel-level classification problem. The most prevalent approach to training semantic segmentation models is full supervision, where the pixel-label annotations are used to train. Via the fully supervised semantic segmentation (FSSS) approach, complicated targets, boundary detection, and localization information can be obtained based on pixel-level segmentation results.

Recently, the success of vision tasks based on the end-to-end training of deep convolutional neural network (DCNN) gives researchers the enthusiasm to explore DCNN-based segmentation. Comparing to the approaches using hand-crafted features, DCNN-based approaches have the capacity to automatically learn the representation through a set of multi-scale feature maps defined in the architecture and a number of convolutional filters.

Although the FSSS approaches have accomplished state-of-the-art performance [1]–[4] on several benchmarks, one difficulty that prevents the FSSS approaches from being widely used is that it needs numerous pixel-wise annotations, which is expensive and time-consuming in reality. Therefore, pixel-wise annotations are regarded as the key to the success of the segmentation tasks. Addressing this problem, the weakly-supervised semantic segmentation (WSSS), only using partial annotations or image-tags to train the segmentation network, is regarded as the solution. In this work, a scribble-based WSSS solution is proposed from which training masks are derived from propagating the semantic information from labelled pixels to unlabelled pixels.

The challenge of the WSSS problem is to use sparse ground truth to obtain compact and accurate segmentation results. Only using weak annotations to train the segmentation network, the results tend to be very sparse, even predict all pixels belong to the background [5]. Nowadays, the popular methods for the WSSS problem are based on Classification Activation Maps (CAMs) [6]. Given training images with only image-level class labels, a classification model is trained, and the class-specific seed areas are obtained during inference. Then, the CAMs are expanded to obtain the Pseudo-Masks [7]–[9], which are used to train a fully supervised semantic segmentation model. However, the CAMs often confuse part of the detected target with background, which is inherited in the segmentation stage.

Scribbles are another kind of economic segmentation ground truth, which is considered as a user-friendly and efficient annotation. Scribble annotations only need the user to drag the cursor in the area of targets, and it need not the whole outline of targets. Thus, scribble annotations avoid inaccurate segmentation cues compared to the CAMs-based approach.

Therefore, this work focuses on using scribble annotations to obtain accurate pixel-wise segmentation for two main tasks. The first one refers to the two industry-related inspection tasks, and the second one is for two multi-category semantic segmentation (MCSS) tasks. Regarding the two inspection tasks, one task is to detect corrosion on the tank surface of vessels (COR task), and another is to detect BioFouling on the vessel's hull (BIO task). In the marine environment, steel materials are prone to severe corrosion and biological contamination, and their damage not only affects the normal operation of vessels but also seriously decreases the lifespan of vessels. Our solution aims at providing an automatic vision-based inspection approach. On the other hand, the inspection tasks only need to detect one category target (corrosion and BioFouling). In order to verify our approach, two multi-category datasets are extracted from Pascal VOC 2012 [10] and CityScapes [11].

For the scribble-based WSSS problem, a closed-form matting (CFM) solution is proposed in this work. Closed-form matting [12] applies the alpha-color model, which establishes

Department of Mathematics and Computer Science (University of the Balearic Islands) and IDISBA (Institut d'Investigacio Sanitaria de les Illes Balears), Palma de Mallorca, Spain; {k.yao, alberto.ortiz, xisco.bonnin}@uib.es

the relationship between the color value and alpha value of each pixel, and it propagates the user annotations to the entire image via minimizing a quadratic cost function. Then, a sparse linear system, where the matting Laplacian matrix is constructed using input image and scribble annotations, is built and solved to obtain the closed-form alpha matte. Considering our problem, the normalized alpha-value from the CFM approach can be seen as the pixel-wise segmentation for one category. Furthermore, a DCNN-based segmentation network trained by the quadratic loss is used to get the optimal alpha solution.

The main contribution of this work are summarized as follows:

- A new loss function (namely closed-form matting loss) aiming at solving the scribble-based WSSS problem is applied to overcome challenges due to ambiguous and sparse annotations.
- A multi-label classification (MLC) model is developed to implement the CFM loss to the MCSS task.
- Some segmentation networks trained with the closed-form matting loss have experimented in this work, and they obtained considerable performance.
- We assess the performance of the matting and segmentation performance on a benchmark comprising two industry-related inspection applications and two MCSS tasks.

This paper is organized as follows: Section II briefly reviews some previous works for the WSSS problem; Section III introduces the methodology of our solution in detail; Section IV reports on the results of a number of experiments aiming at showing the performance of our approach, and Section V concludes this work.

II. Related Work

Numerous FSSS approaches have already been proposed and have achieved impressive performance. Unlike the FSSS problem, which requires plenty of pixel-level ground truth and turns out to be very costly in practice, the WSSS aims at using the partial or image-tag label to get the reasonable, even same segmentation performance as FSSS approaches. Actually, the performance of FSSS approaches is far higher than that of WSSS approaches. Nonetheless, the quality of WSSS methods is impressive, especially considering that learning to segment with a few or without any location-specific supervision is a challenge but brings massive efficiency in the industry scenario. We broadly classified the WSSS problem into three categories according to the class of weak annotations, which are image-tag labels, bounding boxes-based annotations, and scribble-based annotations.

Since image-tag labels completely lack the localization information, most current approaches have two-stage, i.e., generate accurate pseudo ground truth firstly, then use the pseudo labels to train a segmentation network. Typically, SEC [13] proposed three loss functions, called seeding, expansion, and boundary constrain losses, to expand the initial seeds obtained from CAMs [6] and train the FCN-based model. DSRG [7] develops a DCNN-based seed region growing strategy to progressively extend the detected region during training. Their approach starts from the discriminative regions from CAMs, then the region growing module is integrated into a deep segmentation network to obtain a complete segmentation. Then, IRNet [14] is designed for both semantic and instance segmentation. Similarly, their approach uses CAMs as pseudo ground truth, and two branches extending from the backbone are used to predict auxiliary information and the segmentation results.

Some researchers consider that the key factor of the WSSS problem of image-tag labels is to obtain high-quality CAMs. AffinityNet [15] is proposed to learn semantic affinities among adjacent pixels using a limited number of training samples. Additionally, a k-Nearest-Neighbor (kNN) attention pooling layer is applied to decide its k-nearest neighbors, which ensures similar nodes have similar node representation. Stacking kNN attention pooling layers, the AffinityNet can transfer the semantic information from known pixels to their adjacent unknown pixels. In [16], an Attention-based Dropout Layer (ADL) is developed to obtain the entire outline of the target from CAMs. The ADL relies on a self-attention mechanism to process the feature maps. Particularly, during training, this layer hides the most discriminating parts in the feature maps, which induces the network to learn the less discriminating parts while it highlights the informative region of the target to improve the recognition ability. Alternatively, Wei et al. [8] find that varying dilation rates can effectively transfer the discriminative information to non-discriminative regions, promoting the emergence of these regions in CAMs.

Regarding the use of bounding boxes as weakly-supervised annotations for semantic segmentation, in [17], the authors combine two traditional segmentation approaches (GraphCut and Holistically-nested Edge Detection (HED)) algorithms with DCNN to refine the bounding boxes ground truth and make predictions, then the refined ground truth is used to train the network iteratively. An FCN-based network is proposed in [18] to obtain the segmentation results using bounding boxes annotation. Particularly, the authors consider the filling rate, which is the ratio between the size of the target and the size of bounding box annotation, as helpful guidance for obtaining reasonable segmentation performance. Firstly, an FCN and a dense CRF [19] are used to get the segment proposals, then they propose a box-driven class-wise region masking (BCM) and filling rate guided loss (FR-loss) to train the network. Differently, Hsu et al. [20] view the bounding boxes-based WSSS problem as a multiple instance learning (MIL) task. They generate positive and negative bags based on the sweeping lines from the bounding box annotations. The MIL formulation obtains the instance segmentation results by leveraging the tightness property of bounding boxes. Therefore, latent pixel-wise labels, the object instance feature representation, and the segmentation model can be derived simultaneously.

On the other hand, scribbles are another widely used weak annotation for semantic segmentation, which is recognized as one of the most user-friendly ways for labelling. This topic has been explored through graph cuts [21], random walks [22], and weighted geodesic distances [23]. As an improvement, [24] develops two regularization terms based on Normalized Cuts

and dense CRF. In another work [25], a Boundary Perception Guidance (BPG) is proposed, which consists of two components, i.e., prediction and boundary regression. Accordingly, the network has a Prediction Refinement Network (PRN) subnetwork to predict the segmentation results using scribble annotations, and the Boundary Regression Network is used to guide the network to extract edge features using class-agnostic maps as supervision. Besides, Yao et al. [26] apply two kinds of weak annotations (scribbles and pseudo-masks) to train the Attention U-Net [27] for image semantic segmentation. The hierarchical WSSS model consists of two tasks: a. the authors augment the scribble ground truth and train the network to get segmentation results; b. the k-means clustering with scribble annotations is trained in the sub-network, which is used to assist the segmentation task.

Our work falls into the scribble-based WSSS problem. In particular, a deep semantic segmentation network is used to optimize a quadratic loss function to obtain the closed-form solution of a sparse linear system. A specific loss function is designed to propagate the semantic information from the labelled pixels to the unknown label region. Experiments show that our approach is effective in segmenting targets under scribble annotation.

III. Methodology

The methodology of our solution is introduced in this section. Firstly, a brief illustration of the closed-form solution for image matting [12] is presented in Section III-A. Then, the CFM loss is demonstrated in Section III-B. The scribble-based weak annotations used in our solution are introduced in Section III-C. Addressing the MCSS problem, the MLC model is introduced in Section III-D. Finally, we introduce some segmentation networks as the backbone of our approach in III-E, including DeepLabV3+ [28], U-Net [29], and ERFNet [30].

A. Background

For a gray image, the gray-scale value (I_i) of each pixel can be represented as,

$$I_i = \alpha_i F_i + (1-\alpha_i) B_i \tag{1}$$

where F represents the foreground, B represents the background, and α denotes the alpha channel. Redoing (Eq. 1), we can get the expression as follow.

$$\alpha_i = \frac{1}{F_i - B_i} I_i + (-\frac{B_i}{F_i - B_i}) \tag{2}$$

Some assumptions on the nature of F_i, B_i and α_i are needed. Assume that both F_i and B_i are approximately constant over a small window around each pixel. This assumption allows us to rewrite (Eq. 2), expressing α as a linear function of image I in Eq. 3,

$$\alpha_i \approx a_i I_i + b_i, \forall i \in w \tag{3}$$

where $a_i = \frac{1}{F_i - B_i}$, $b_i = -\frac{B_i}{F_i - B_i}$, and w is a small image window, whose size is 3×3 as usual. So the relation suggests finding α, a_i and b_i that minimizes the cost function as below,

$$J(\alpha, a_i, b_i) = \sum_{j \in k} \left(\sum_{i \in w_k} (\alpha_i - a_j I_i - b_j)^2 + \epsilon a_j^2 \right) \tag{4}$$

where w_k is a small window around pixel j. The cost function in Eq. 4 includes a regularization term on a. One reason for this term is numerical stability [12].

The cost function is quadratic in α, a_i, and b_i, with $3N$ unknowns for an image with N pixels. Through several derivations, a_i and b_i can be removed from Eq. 4, leaving us with a quadratic cost in only N unknowns: the α values of the pixels,

$$J(\alpha) = \alpha^T L \alpha \tag{5}$$

where, L is the Matting Laplacian matrix, whose size is $N \times N$. The (i,j)th entry is,

$$\sum_{k|(i,j) \in w_k} \left(\delta_{ij} - \frac{1}{|w_k|} \left(1 + \frac{1}{\frac{\epsilon}{|w_k|} + \sigma_k^2} (I_i - \mu_k)(I_j - \mu_k) \right) \right) \tag{6}$$

where, w_k presents a small window of 3×3 pixels. Since we place a window around each pixel, the window, μ_k and σ_k^2 are the mean and variance of the intensities in the window w_k around k, and $|w_k|$ is the number of pixels in this window. δ_{ij} is the Kronecker delta as shown below.

$$\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \tag{7}$$

The alpha value for color image, a BGR image for instance, also can be obtained by optimizing the quadratic form loss as Eq. 5. The (i,j)th entry of the Matting Laplacian Matrix is,

$$\sum_{k|(i,j) \in w_k} \left(\delta_{ij} - \frac{1}{m} \left(1 + (I_i - \mu_k)^T \left(\Sigma_k + \frac{\epsilon I_3}{m} \right) (I_j - \mu_k) \right) \right) \tag{8}$$

where, I_3 is a 3×3 identity matrix, μ_k is the 3×1 mean value of colors in window w_k, Σ_k is the 3×3 covariance matrix of the intensities in window w_k.

To provide the user with more flexible control over the output, the scribble annotations are used for the constraint priors. In order to extract an alpha matte matching the user's constraints, the optimized problem has become to find the optimal α^* in Eq. 9.

$$\alpha^* = \underset{\alpha}{\operatorname{argmin}} \left[\alpha^T L \alpha + \lambda (\alpha - S)^T D_s (\alpha - S) \right] \tag{9}$$

where, D_s is a diagonal matrix whose diagonal elements are one for constrained pixels and zero for all other pixels, S is the vector containing the specified alpha values for the constrained prior and zero for all other pixels, and λ is a large number (100 in our experiments).

As can be seen, the loss function in Eq. 9 is a quadratic in α, the optimal solution can be found by *Lagrange multiplier*. In the end, this is equivalent to solving the following sparse linear system in Eq. 10.

$$(L + \lambda D_s)\alpha = \lambda S \tag{10}$$

B. Closed Form Matting Loss Function

As known in Eq. 9, the optimal solution $\alpha*$ can be obtained by solving the sparse linear system in Eq. 10. Inspired by the success of DCNN-based models, we consider using a DCNN to predict the α in Eq. 10, and change the original optimized problem to a DCNN-based learning problem. Note that the matting problem in [12] needs two inputs (image and scribble annotations) to obtain the optimal solution. After changing to a learning problem, the DCNN-based model only needs one input image to obtain the final solution with less running time during inference. Therefore, our DCNN-based solution is more efficient and requires fewer inputs.

Specifically, we use a Mean Square Error (MSE) loss function with a DCNN-based model to obtain the approximate optimal solution. The loss function are modified based on Eq. 10, and it can be expressed in Eq. 11.

$$L_{\text{cfm}}(\alpha, \theta) = ||(L + \lambda D_s)\alpha(\theta) - \lambda D_s S||^2 \tag{11}$$

where, θ represents the weights of the segmentation model, and α denotes the network prediction. The definitions of L, D_s, and S are the same as in Eq. 9, which can be obtained from the input image and constraint priors (scribble annotations).

Let $A = L + \lambda D_s$ and $B = \lambda D_s S$, so the loss function in Eq. 11 can be simplified as follow.

$$L_{\text{cfm}}(\alpha, \theta) = ||A\alpha(\theta) - B||^2 \tag{12}$$

In Eq. 12, α is the prediction of one category and w_c denotes the weight for every category, which is used to solve the imbalance problem between different categories. For the multi-category problem (assuming C categories), the loss function can be expressed as:

$$L_{\text{cfm}}(\alpha_c, \theta, w_c) = \frac{\sum_{c=1}^{C} ||w_c(A_c\alpha_c(\theta) - B_c)||^2}{C}, \quad c \in C \tag{13}$$

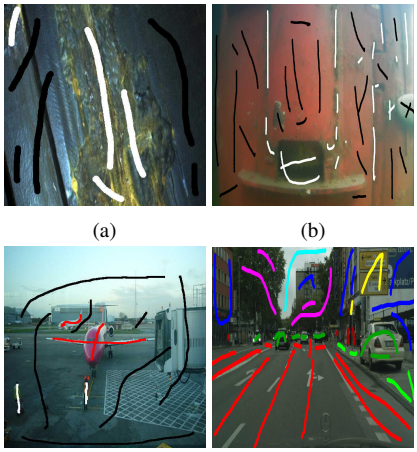
where $A_c = L + \lambda D_{s,c}$, $B_c = \lambda D_{s,c}$.

C. Scribble Annotations

In our system, we apply the scribbles-based weak annotations as the prior constraints. Let α_{gt} be the scribble annotations for category c, which can be formally defined in Eq. 14,

$$\alpha_{gt,c}(x, y) = \begin{cases} 0, \text{background} \\ 128, \text{unknown} \\ 255, \text{foreground} \end{cases}, \quad c \in C \tag{14}$$

where (x, y) indicates the coordinates of pixel in the image. Intuitively, the scribble annotations separate the semitransparent regions from the opaque foreground and background. In order to obtain the optimal α^*, the matting approach naturally divided two steps: a. decide pixels to belong foreground or background for labelled pixels; b. compute the alpha value for a pixel locates in the unknown regions. Reminiscent the WSSS task, whose target is to divide the image into discrete parts according to the semantic information, the partial matting



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

Fig. 1. Samples of weak annotations, from more to less informative: (a) and (b) are samples from the inspection tasks, and only black and white scribbles are used due to the binary segmentation task; (c) and (d) come form the MCSS tasks, and different color scribbles are used to distinguish the category of target.

alpha (scribble) annotation is helpful for the segmentation task in two aspects. On the one hand, propagating the α value for unlabelled pixels can significantly relieve the burden of the network to predict the extracted label for these pixels. On the other hand, the labelled pixels can provide prior knowledge for the network and also enhance the regression performance of the network for these labelled pixels. Besides, if the user annotations contain minor errors, the propagation mechanism would like to correct them.

Figure 1 shows four example pictures of our tasks. As can be seen, the unknown region in the scribble annotations is vast and erroneous, while only a few pixels are labelled. Rightfully, directly using scribble annotations to train segmentation network cannot achieve satisfactory performance, even does not converge. However, our approach can obtain precise matte and segmentation results, thus perfectly solves our problems.

D. Multi-Label Classification

In order to solve the Multi-Label problem, a simple Multi-Label Classification (MLC) model is developed, which consists of an average pooling layer, two Fully-Connected (FC) layers, and a Sigmoid activation function. The MLC model is connected with the encoder, as shown in Fig. 11. During training, we employ a Cross-Entropy Loss to train the MLC model. Regarding the classification ground truth, the scribble annotation can provide the information of the targets' category appeared in the image.

In the end, the loss function (L) of multi-label problem is shown as follow,

$$\begin{aligned} L &= L_{\text{cfm}} + \beta L_{\text{MLC}}, \\ L_{\text{MLC}} &= \sum_{c}^{C} y_c \log(p_c) + (1 - y_c) \log(1 - p_c) \end{aligned} \tag{15}$$

where C represents the number of categories, β is the trade-off constant, y_c indicates the ground truth and p_c denotes the output of MLC model.

E. Segmentation Backbones

For the task of semantic segmentation, the spatial pyramid pooling module and auto encoder-decoder structure are widely used in deep neural networks. The spatial pyramid pooling module can encode multi-scale contextual information by probing the incoming features with filters or pooling operations at multiple rates and multiple effective fields-of-view. While the auto decoder model makes full use of the multi-scale and rich semantic information features from DCNN, which is helpful to obtain an accurate target profile. In this section, we introduce different segmentation networks as the backbones in our solution in order to validate the effectiveness of our approach.

1) DeepLabV3+: DeepLabV3+ [28] extends the DeepLabV3 [3] by adding a practical decoder module to refine the segmentation results. The DeepLabV3+, applied in this work, has the ability to provide good segmentation detection, especially in the domain of scribble-based weakly supervised semantic segmentation problems. The architecture of our approach can be seen in Fig. 2. Here, a brief review of DeepLabV3+ is presented.

a) Atrous Convolution: The atrous convolution (dilated convolution) introduces a new parameter to the standard convolution called the "dilate rate", which is used to explicitly control the resolution of feature maps of DCNN. Compared to the standard convolution, the atrous convolution can obtain a larger receptive field, which can improve the segmentation performance and also increase the localization accuracy. On the other hand, the network can capture multi-scale contextual information by different dilated factors. Thus, the atrous convolution can provide the necessary information that is needed to improve the image segmentation performance [1], [31].

Considering two-dimensional inputs, for each location i on the output y and a filter w, atrous convolution is applied over the input feature map x as in Eq. 16.

$$y\left[i\right] = \sum_{k} x\left[i + r \cdot k\right] \;\; w\left[k\right] \tag{16}$$

where the dilated rate r directly affects the stride of convolution sampling. Furthermore, the standard convolution is considered as a special case, where $r = 1$. Via changing the r value, the receptive field of convolution can be changed adaptively. As shown in Fig. 3, using different dilated rate values in convolution operation, the receptive field changes significantly. Besides, stacking atrous convolution makes the receptive field growing exponentially in the DCNN model.

b) Depthwise Separable Convolution: The Depthwise Separable (DS) convolution [32] is the combination of Depthwise and Pointwise convolution, which can drastically reduce computation complexity. Specifically, the DS convolution decomposes the standard convolution into Depthwise convolution, whose number of output channels equals the number of input channels, and then Pointwise convolution (1×1 convolution) is applied on the output of Depthwise convolution to obtain the target number of output channels.

In DeepLabV3+, the authors combine the atrous convolution with DS convolution, namely atrous separable convolution, and prove that atrous separable convolution significantly reduces the computation complexity of the proposed model while maintaining better performance [3].

c) Atrous Spatial Pyramid Pooling: The Atrous Spatial Pyramid Pooling (ASPP) model is proposed in [1], and it is inspired by spatial pyramid pooling [33], [34], which can effectively improve the accuracy and efficiency for classifying regions of an arbitrary scale. The ASPP model in DeepLabV3+ applies three atrous convolutions with different atrous rates to capture multi-scale information. Besides, an average pooling and a 1×1 convolution layer with 256 filters are applied on the input feature maps to obtain the image-level features. Therefore, the ASPP model consists of (a) a 1×1 convolution and three 3×3 convolutions with different dilated rates (6, 12, 18) and (b) a global average pooling. In the end, the resulting features from all branches are concatenated in the channel and pass through another 1×1 convolution (256 filters and batch normalization). The architecture of the ASPP model can be seen in Fig. 2.

d) Encoder-Decoder Architecture: The DeepLabV3+ is developed based on the DeepLabV3, taking advantage of atrous convolution to extract features of the DCNNs at arbitrary resolution and with a large receptive field, and to augment the ASPP model aimed at using multi-scale features by different dilated rates. The DeepLabV3 applies a naive decoder module, which bilinearly upsamples the features by a factor of 16. However, this work lacks the accuracy of the target's boundary in the semantic segmentation task. To address this problem, DeepLabV3+ proposes a simple but effective decoder module. As shown in Fig. 2, the last feature maps of the encoder are upsampled by a factor of 4, and then concatenated with the corresponding low-level feature maps [2]. After, the output is obtained through a few 3×3 convolutions to refine the feature maps, followed by another simple bilinear upsampling by a factor of 4. Herein, the proposed encoder-decoder structure is implemented to take the place of the single-applied decoder structure in [3] and improve the target boundary's segmentation performance.

e) Modified Aligned Xception: The Xception model [32], which is employed as network backbone (denoted as X-65) in object detection [35] and segmentation [28] task, has been proved that it is useful to push the performance on the large scale dataset, such as MS-COCO [36], PASCAL VOC 2012 [10]. Therefore, we also employ the Aligned Xception model in our work.

Specifically, some changes based on [35] need to be conducted, namely (1) similar to MRSA's work, the depth of Xception model is increased in this work apart from the structure of entry flow considering fast computation and memory efficiency; (2) all of the max-pooling layers in Xception model are replaced by depthwise separable convolution, where combines stride and dilated rate to obtain the arbitrary resolution feature maps; (3) the batch normalization and ReLU are used behind the 3×3 depthwise convolution.

2) U-Net: The U-Net [29] is developed for BioMedical image segmentation, and its architecture is illustrated in Fig. 4. The U-Net consists of two paths. The first path is named as contraction path (also called the encoder), which is used

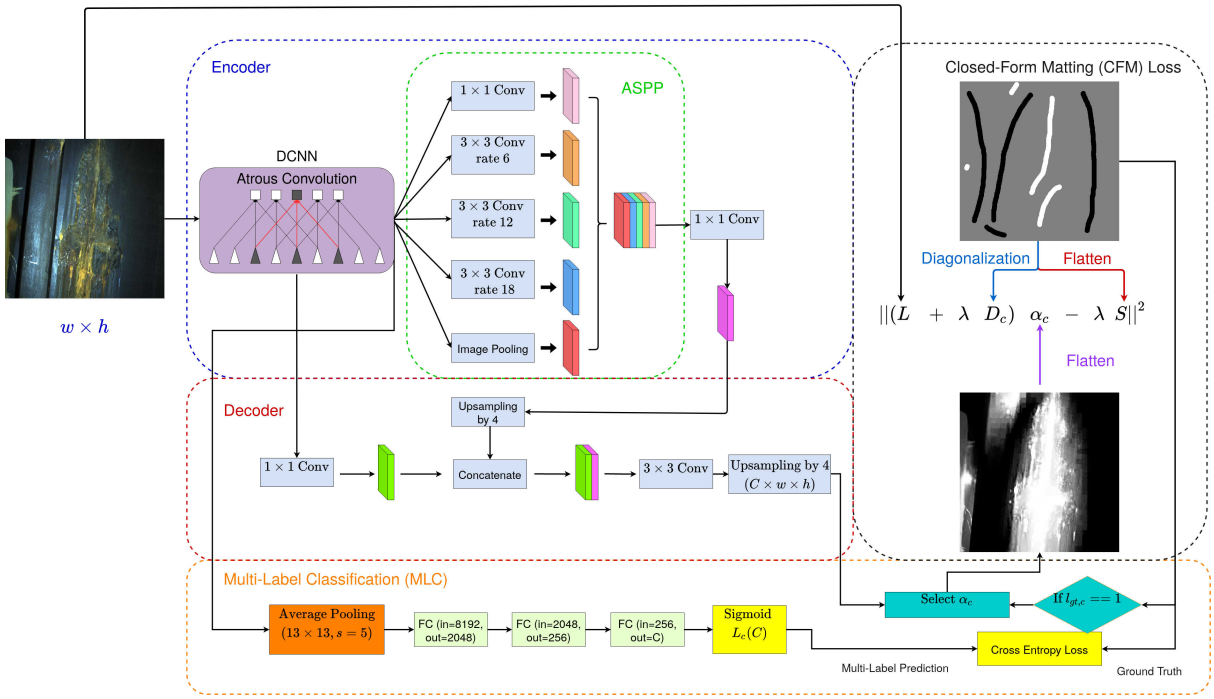


Fig. 2. Training schematic diagram of DeepLabV3+ with the closed form loss. The encoder model is employed to extract multi-scale contextual information by using atrous convolution, then the ASPP model is used to integrate multi-scale information. A simple but effective decoder model is proposed to refine the segmentation results. The MLC model is connected to the encoder, which is used to obtain the category information of targets in the image. In the end, our CFM loss is used to train the network.

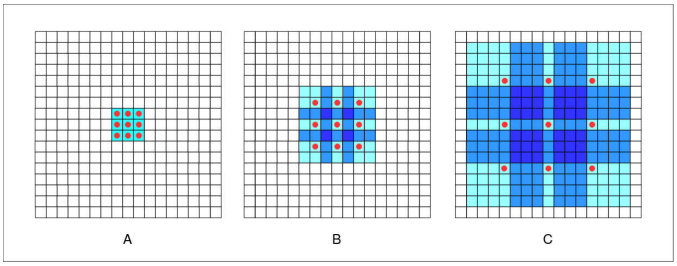


Fig. 3. A diagrams illustration of the atrous convolution. In this figure. only 9 red points for a 3×3 kernel are involved in the convolution operation, and other points in the receptive field are ignored. A is the standard convolution, where the r is 1, its receptive field is only 3×3; the r in B is 2, and its receptive field is 7×7; while C is the 4-dilated convolution, and its receptive field is 15×15.

to capture the context from the input image. The encoder is a stack of convolutional and pooling layers. The second path is the symmetric expanding path (also named the decoder) which is used to enable precise location combined with the feature maps of the encoder. Therefore, U-Net is an end-to-end fully convolutional network, and it achieves outstanding performance for medical image segmentation.

3) ERFNet: ERFNet [30] is a lightweight segmentation network, which proposes a factorized residual (FR) layer to remain efficient while retaining remarkable segmentation performance. The residual layer [37] has been proved that it can facilitate training and significantly reduce the performance degradation due to stacking a large number of convolutional layers. Specifically, the non-bottleneck in ResNets gains the accuracy for the increased depth network [37]–[39]. Inspired by the Non-bottleneck, a modified residual module is proposed using convolutions with 1D filters, namely non-bottleneck-1D module, as shown in Fig. 5. By leveraging the decomposition in Fig. 5 [b], the non-bottleneck-1D can accelerate running time and reduce the parameters of the non-bottleneck module.

The architecture of ERFNet follows a prevail encoder-decoder structure. Inspired by ENet [40], a complicated but efficient encoder is designed to gather context from the input image. Particularly, every down-sample block of ERFNet performs down-sampling by concatenating outputs of a convolutional layer using 3×3 kernels with stride 2, and a max-pooling layer. After down-sampling, some non-bottleneck-1D modules with different dilated rates [41] are applied to improve the segmentation performance. Similar to the design of ENet, a simple decoder is used to up-sample the encoder's output to match the input resolution. For every up-sample block, a deconvolution layer with stride 2 is adopted. The architecture of ERFNet is fully depicted as in Table I.

IV. Experiments and Discussion

In this section, we conduct experiments and illustrate the performance of our approach on our two inspection tasks and two MCSS tasks. In Section IV-A, the experimental environments, including the dataset, the evaluated metrics, and the hyper-parameters of experiments, are introduced. Then, we evaluate the matting performance for the inspection tasks in Section IV-B, and the segmentation performance for inspection tasks is illustrated in Section IV-C. For the MCSS tasks, the segmentation performance is compared in Section IV-D. In the end, some examples of segmentation results for our two tasks are shown in Section IV-E.

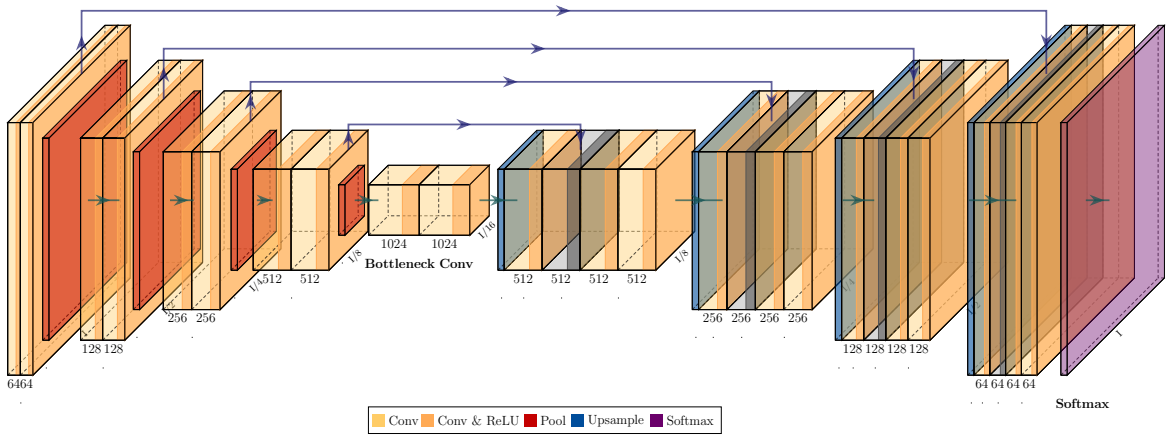


Fig. 4. The architecture of U-Net. In this figure, *Conv* represents convolutional layer with 3×3 kernel; *Conv & ReLU* indicates convolutional layer connected with *ReLU* activation; *Pool* layer is used to decrease the dimension of the input feature; *Upsample* is used to increase the dimension of the input feature; *Softmax* activated function is used to get the segmentation for every category.

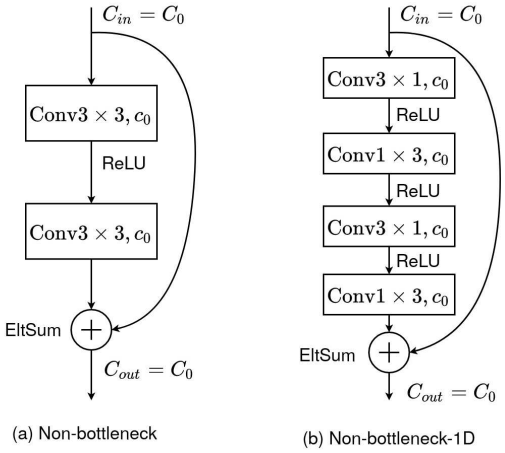


Fig. 5. The illustration of (a) Non-bottleneck in [37] and (b) Non-bottleneck-1D in [30]. In this figure, C represents the channels of feature map. In the convolution block, $(d_1 \times d_2,\ c)$ indicates the kernel size (d_1, d_2) and the output channels (c) of feature maps. *EltSum* denotes element-wise sum.

TABLE I
A DETAILED DEMONSTRATION FOR THE ARCHITECTURE OF ERFNET. *OUT-C* REPRESENTS THE OUTPUT CHANNELS, *OUT-DIM* INDICATES THE DIMENSION OF OUTPUT FEATURE MAPS, AND *C* DENOTES THE NUMBER OF CATEGORY.

	Type	OUT-C	OUT-DIM
Encoder	down-sample block	16	512×256
	down-sample block	64	256×128
	$5\times$ non-bottleneck-1D	64	256×128
	down-sample block	128	128×64
	non-bottleneck-1D (dilated 2)	128	128×64
	non-bottleneck-1D (dilated 4)	128	128×64
	non-bottleneck-1D (dilated 8)	128	128×64
	non-bottleneck-1D (dilated 16)	128	128×64
	non-bottleneck-1D (dilated 2)	128	128×64
	non-bottleneck-1D (dilated 4)	128	128×64
	non-bottleneck-1D (dilated 8)	128	128×64
	non-bottleneck-1D (dilated 16)	128	128×64
Decoder	up-sample block	64	256×128
	$2\times$ non-bottleneck-1D	64	256×128
	up-sample block	16	512×256
	$2\times$ non-bottleneck-1D	16	512×256
	up-sample block	C	1024×512

A. Experiment Settings

a) Datasets: The dataset from the COR task consists of 241 images in total, three-fourths of which are designated for training and the rest for testing. Regarding the dataset of the BIO task, it contains 195 images, and the same strategy is adopted for splitting into the training and validation sets. Both datasets are obtained from real industry scenes. During training, an online data augmentation strategy, including rotation, scaling, and random cropping, is used to increase the diversity of our training dataset.

In order to verify the effectiveness of our solution for the MCSS task, two subsets are extracted from Pascal VOC 2012 and CityScapes, respectively. The VOC subset consists of 453 images and has 6 categories, which are plane, bike, person, car, dog, and cat, while the CityScape subset contains 450 images with 9 categories (road, building, sidewalk, traffic sign, plant, sky, person, car, and bike). For both datasets, we select 400 images to train the network, and the rest is used for validation.

To obtain the scribble annotations, a color brush is used to partially mark the pixels of target and background, as shown in Fig. 1. Other pixels, whose grayscale values are 128, represent unknown, which need the network to learn its label and propagate the information from labelled pixels.

b) Implementation Details: All experiments have been implemented using the Pytorch framework running in a PC fitted with an NVIDIA GeForce RTX 2080 Ti GPU, a 2.9 GHz 12-core CPU with 32 GB RAM, and Ubuntu 64-bit. For experiments using the DeepLabV3+ model, we adopt the same training details for three different backbones (ResNet101, Xception, and MobileNet), i.e., initializing the backbones using the ImageNet [42] pre-trained model, resizing the input images to 513×513 dimensions for the inspection tasks and 321×321 for the MCSS tasks, optimizing the network via the Adam algorithm, setting the initial learning rate as 0.001. In

the case of U-Net as the segmentation backbone, we resize the input images to 512×512 dimensions, while the input images are resized to 320×160 in the case of ERFNet. Regarding λ in (11), we set it as 100, which is the same as in [12].

c) Evaluation Metrics: For quantitative evaluation of our approach, we evaluate the performance of our approach in two aspects: a. the accuracy of the closed-form solution; b. the image segmentation performance.

As mentioned in section III-B, our approach essentially uses a neural network to obtain the approximate closed-form solution of a sparse linear system. Therefore, we use three metrics to evaluate our approach. Regarding the Matte ground truth, we select the results of the CFM approach. The metrics are explained in the following.

- The mean Absolute Differences (mAD) can be formally stated as follows: give the alpha of our approach α_i for a total of N pixels, and the solution of Closed-Form Matting (CFM) approach α_i^*, it is defined as Eq. 17.

$$\text{mAD} = \frac{\sum_{i \in N} |\alpha_i - \alpha_i^*|}{N} \tag{17}$$

  The Mean Square Error (MSE) is to compute the mean Euclidean distance between the predicted α_i and the solution of CFM approach α_i^*. It can be expressed as follows.

$$\text{MSE} = \frac{\sum_{i \in N} (\alpha_i - \alpha_i^*)^2}{N} \tag{18}$$

- The Gradient Error (GE) is to calculate the gradient difference between the alpha matte α_i and the output α_i^* of the CFM approach, which is defined in Eq. 19. Here, ∇ means the gradient, which can be obtained by convolution operation using a first-order Gaussian derivative filter. q is a hyper-parameter, and it is set to 2 in our experiments.

$$\text{GE} = \sum_{i \in N} (\nabla \alpha_i - \nabla \alpha_i^*)^q \tag{19}$$

To obtain the segmentation mask from the matting results, four thresholds (0.4, 0.5, 0.6, 0.7) are set for the matting results α. Specifically, the segmentation results of category c ($p_{seg,c}$) are obtained as follows,

$$p_{seg,c} = \begin{cases} 1, & \alpha \geq \gamma \\ 0, & \alpha < \gamma \end{cases} \quad \gamma \in [0.4, 0.5, 0.6, 0.7] \tag{20}$$

In order to evaluate the performance of segmentation, some common metrics are considered, including:

- The Intersection Over Union (IOU) is to compute the ratio of intersection and union between the prediction and ground truth. Formally, let n_{ij} be the number of pixels of class i fall into class j, the IOU is defined as in Eq. 21,

$$\text{IOU} = \sum_i \frac{n_{ii}}{\sum_j n_{ij} + \sum_j n_{ji} - n_{ii}} \tag{21}$$

- The Recall (Rec) and Precision (Prec) also are computed to evaluate the performance of our approach, which can be expressed as follows:

$$\begin{aligned} \text{Rec} &= \sum_i \frac{TP_i}{TP_i + FN_i} = \sum_i \frac{TP_i}{T_i} \\ \text{Prev} &= \sum_i \frac{TP_i}{TP_i + FP_i} = \sum_i \frac{TP_i}{P_i} \end{aligned} \tag{22}$$

  where TP_i, FP_i and FN_i are respectively the true positives, false positives and false negatives for class i and T_i and P_i are, respectively, the number of positives in the ground truth and the number of predicted positives, both for class i.
- the F1 score is used as the the harmonic mean of precision and recall:

$$\text{F1} = \frac{2 \cdot \text{mPrec} \cdot \text{mRec}}{\text{mPrec} + \text{mRec}} \tag{23}$$

In the case of the COR task, we make use of fully supervised masks/ground truth in order to be able to report accurate segmentation metrics. This ground truth has been manually generated only for this purpose. Regarding the BIO task, since the photos are taken in the underwater environment, it is difficult and expensive to obtain the fully supervised pixel-wise ground truth. Therefore, we obtain the ground truth from the matting results of the CFM approach by setting thresholds, which is only for the purpose of evaluation.

Regarding the MCSS tasks, the IOU is obtained for every category. Then the mIOU is attained by mean of every category IOU value.

B. Evaluation for Matting Performance

Compared to the CFM algorithm, our approach implements the same loss function. Differently, a segmentation network is used to obtain the closed-form solution. Therefore, our approach eliminates redundant iterative optimized procedures. In this section, variants of segmentation networks are trained and compared with the results of the CFM algorithm.

There are already many image segmentation networks, which have achieved good segmentation performance. In this work, we select to use DeepLabV3+ [28], U-Net [29], which obtains excellent segmentation performance for natural and medical images, and ERFNet [30], which applies a small scale network to obtain good segmentation performance for the CityScape benchmark [11].

We respectively select one example from our two inspection tasks to compare the values of solution in Fig. 6. Comparing the distribution of the solutions from different segmentation networks, it can be seen that different solutions' distributions have similar a shape. Especially for the COR task (the second row in Fig. 6), using DeepLabV3+ with Xception backbone can obtain additional mattes, which have a similar character to corrosion. In our case, these additional mattes can be seen as suspicious targets.

In Table II, we compute the MSE, mAD, and GE to evaluate how close the solution of our approach is to the solution of the CFM algorithm. Particularly, applying different backbones in the DeepLabV3+ module for our two tasks, the values of MSE, mAD, and GE is slightly different on both training and validation set. The performance of U-Net for the BIO task

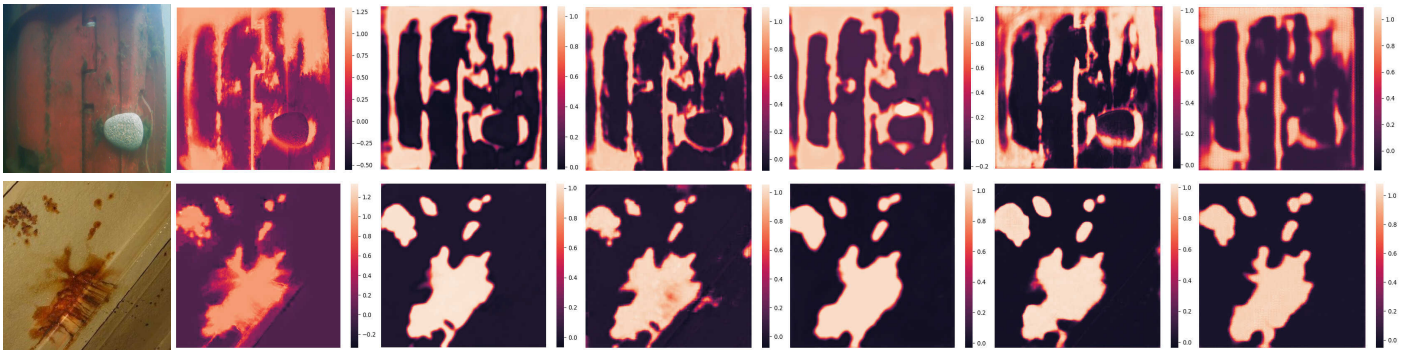


Fig. 6. The heat map of matting results: the first row shows the example from the BIO task, and the second row represents the example form COR task; the first column shows the input image, the matting results from CFM algorithm is shown in the second column; from the 3- to 7-th rows, the matting results from Res101-, Xception-, MobileNet-based DeepLabV3+ network, U-Net, and ERFNet are shown.

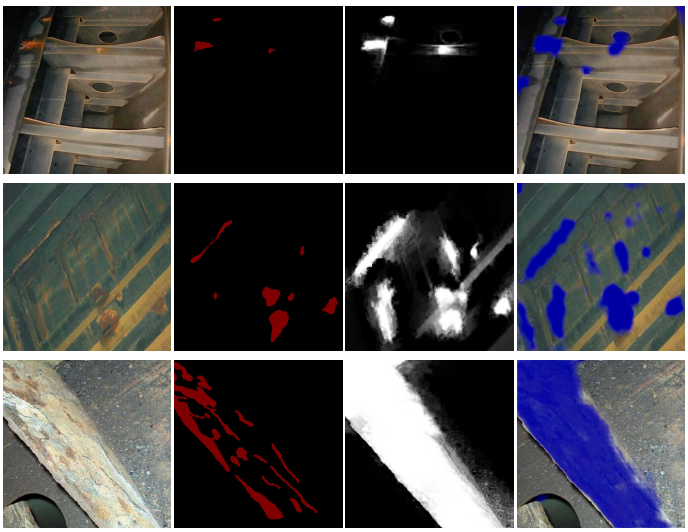


Fig. 7. Examples of matting results and ground truth of the COR task.

is slightly lower than the performance of the DeepLabV3+ network, while its performance for the COR task is close to DeepLabV3+. Notice that, for the small-scale network, such as the ERFNet (6.87MB parameters) and DeepLabV3+ (MobileNet as the backbone, 23.6 MB parameters), our approach also can achieve substantial performance as shown in Table II.

On the other hand, observing the values of metrics in Table II, different segmentation networks, that are trained by our loss function, obtain similar matting performance for our two inspection tasks. Therefore, the CFM loss can be used in different segmentation backbones.

C. Segmentation Results of Inspection Tasks

To evaluate the segmentation performance of our approach, we measure the metrics on the two inspection tasks. For the BIO task, it is expensive in cost and time to obtain the pixel-level ground truth. Therefore, we obtain its ground truth by thresholding the CFM results, where the threshold is set to 0.5. Regarding the COR task, the pixel-level ground truth is used to evaluate the segmentation performance. Table III displays the metrics of segmentation performance. In order to obtain the segmentation results, four thresholds (from 0.4 to 0.7) are set for the matting results.

As for the COR task, compared to the mIOU and F1 score, our approach achieves the best performance when the threshold is set to 0.7. When the threshold is 0.4, the best recall is obtained. Since our approach propagates scribble annotations to the entire image by minimizing the quadratic cost function, it can provide the suspicious regions according to how similar to the features of scribble annotations. On the other hand, these suspicious regions are not labelled due to their small scale and low hazard, but it is useful for us to analyze the inspection results. As shown in Fig. 7, the matte results provide useful suspicious corrosion regions, that are not labelled in the ground truth. Therefore, it can explain the reason that precision values for the COR task are low. Conversely, our approach achieves high recall values. In all, our approach can provide reliable COR results.

As for the BIO task, three DeepLabV3+ models obtain outstanding segmentation performance than U-Net and ERFNet. Compared to metrics of DeepLabV3+ models, it is clear that when the threshold is set to 0.7, DeepLabV3+ with ResNet101 and Xception attains the highest recall, mIOU, and the highest precision, F1 score, respectively. For the MobileNet-based network, which only has 23.6 MB parameters, it obtains 0.8359 mIOU and 0.8721 F1 scores. Thus, even using a small-scale segmentation network, our approach can provide reasonable inspection results.

D. Segmentation Results of MCSS Tasks

We conduct experiments to evaluate the effectiveness of our approach for MCSS tasks, and all results are evaluated on the two subsets of Pascal VOC and CityScapes benchmarks. For comparison, the upper bound of our method corresponds to the configuration using full masks and the cross-entropy loss (L_{ce}) for training. The partial cross-entropy loss (L_{pce}) with scribble annotations is used as the lower bound of our approach. Besides, we try to use a combination loss of L_{pce} and L_{cfm} to compare the segmentation performance.

The experimental results are reported in Table IV for the VOC subset, and in Table V for the CityScape subset. Different

TABLE II
METRICS FOR THE DIFFERENT EXPERIMENTS PERFORMED, VARYING THE SEGMENTATION NETWORK. MSE STANDS FOR THE MEAN SQUARE ERROR, MAD INDICATES THE MEAN SUM OF ABSOLUTE DIFFERENCE, AND GE REPRESENTS THE GRADIENT ERROR.

Dataset	Network (backbone)	MSE	mAD	GE
BioFouling (training set)	DeepLabV3+ (ResNet-101)	0.0286	0.0822	2.1313
	DeepLabV3+ (Xception)	0.0266	0.0751	2.2626
	DeepLabV3+ (MobileNet)	0.0296	0.0811	2.0556
	U-Net	0.0599	0.1404	2.9092
	ERFNet	0.0480	0.1128	2.1250
BioFouling (validation set)	DeepLabV3+ (ResNet-101)	0.0302	0.0837	1.7876
	DeepLabV3+ (Xception)	0.0309	0.0821	1.9361
	DeepLabV3+ (MobileNet)	0.0349	0.0889	1.6987
	U-Net	0.0625	0.1446	2.7250
	ERFNet	0.0501	0.1151	1.8171
Corrosion (training set)	DeepLabV3+ (ResNet-101)	0.0368	0.0829	1.9733
	DeepLabV3+ (Xception)	0.0443	0.1017	2.1341
	DeepLabV3+ (MobileNet)	0.0341	0.0793	1.9124
	U-Net	0.0467	0.0993	2.3060
	ERFNet	0.0407	0.0871	2.2085
Corrosion (validation set)	DeepLabV3+ (ResNet-101)	0.0399	0.0893	1.9401
	DeepLabV3+ (Xception)	0.0519	0.1146	2.1266
	DeepLabV3+ (MobileNet)	0.0363	0.0845	1.8927
	U-Net	0.0457	0.0981	2.4294
	ERFNet	0.0432	0.0925	2.1210

from the inspection task, the threshold (0.5) is used to obtain the segmentation results from the matting results of every category.

In Table IV, the comparison reflects that our approach has the outstanding performance than the L_{pce} method, and the IOU of Dog class (0.7459) is higher than the IOU of fully-supervised approach (0.7250). For the network trained by the combinational loss function, it attains a lower segmentation performance than that of network trained by L_{cfm}.

For the subset of CityScape, training with our loss function obtains higher mIOU than the network training through L_{pce} loss, as shown in Table V. Compared to the fully-supervised approach, the mIOU is 0.7136, which is 0.0731 higher than our approach. However, for the relatively big target, such as Road, Building, Sidewalk, Plant, Sky, and Car, our approach has a similar performance to the fully-supervised approach. Same as the experiment on the VOC subset, L_{pce} has a negative effect on the segmentation performance in the case of combinational loss function.

E. Experimental Results Visualization

In Fig. 8 and 9, some samples from BioFouling and COR tasks are shown, respectively. For one input image, the first row displays the input image firstly, then the scribble annotation is shown, and the rest in the first row displays the segmentation network. For both tasks, the segmentation results are obtained by setting the threshold to 0.7. The second row firstly shows the evaluation ground truth; secondly, the matting results of the CFM algorithm are shown. The rest of the image in the second row are the matting results from our solution.

As shown in Fig. 8 and 9, our approach can obtain similar matting results to the CFM algorithm, which provides considerable segmentation results by thresholding.

For the MCSS problems, the segmentation results on the subsets of VOC and CityScape, which are obtained by thresholding on the matting results, are shown in Fig. 10 and 11, respectively. As can be observed, our approach achieves a considerable matting performance for every category. At the end, we set the threshold of 0.5 to the matting results on each category to get the final segmentation results. As can be seen in Fig. 10 and 11, our approach gets a good segmentation performance for different MCSS tasks.

In conclusion, the network, trained by our CFM loss function with scribble annotations, can obtain a reliable segmentation performance for the Inspection and MCSS tasks.

V. CONCLUSION

This paper explores a simple but effective approach to obtain reliable segmentation results for two inspection tasks and two multi-category semantic segmentation tasks only using scribble annotations. In particular, a matting-based loss function, namely closed-form matting (CFM) loss, is proposed to propagate the user annotations to the unknown label regions. In this work, we apply our loss function to train some segmentation networks (DeepLabV3+, U-Net, and ERFNet), and they achieve reasonable matting and segmentation results. For the MCSS problem, an extra MLC model is added in the segmentation network, which is used to predict objects' categories in the image. Our solution is simple but effective to obtain reliable segmentation performance.

REFERENCES

[1] 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, no. 4, pp. 834–848, 2017.

[2] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 447–456.

TABLE III

LIST OF SEGMENTATION METRICS ON THE VALIDATION SET FOR THE TWO INSPECTION TASKS. *Network(backbone)* REFERS THE SEGMENTATION NETWORK INVOLVED IN OUR SOLUTION; *#TH* STANDS FOR THE THRESHOLD TO GET THE SEGMENTATION RESULT BASED ON THE MATTING RESULTS; *mIOU* REPRESENTS THE MEAN INTERSECTION OVER UNION; *Rec*, *Prec*, AND *F1* INDICATES THE RECALL, PRECISION, AND F1 RESPECTIVELY. BEST PERFORMANCE IS HIGHLIGHTED IN BOLD.

Dataset	Network (backbone)	#TH	mIOU	Rec	Prec	F1
BioFouling	DeepLabV3+ (ResNet-101)	0.4	0.8264	**0.9285**	0.8115	0.8660
		0.5	0.8345	0.9139	0.8338	0.8720
		0.6	0.8393	0.8972	0.8545	0.8753
		0.7	**0.8403**	0.8754	**0.8759**	**0.8756**
	DeepLabV3+ (Xception)	0.4	0.8262	**0.9164**	0.8158	0.8685
		0.5	0.8338	0.8973	0.8414	0.8761
		0.6	**0.8367**	0.8760	0.8643	0.8804
		0.7	0.8342	0.8488	**0.8866**	**0.8809**
	DeepLabV3+ (Mobilenet)	0.4	0.8193	**0.9453**	0.7851	0.8577
		0.5	0.8284	0.9319	0.8072	0.8650
		0.6	0.8340	0.9162	0.8281	0.8699
		0.7	**0.8359**	0.8957	**0.8498**	**0.8721**
	U-Net	0.4	0.7134	**0.8273**	0.7101	0.7642
		0.5	**0.7191**	0.7871	0.7436	**0.7647**
		0.6	0.7161	0.7420	0.7737	0.7575
		0.7	0.7024	0.6836	**0.8043**	0.7390
	ERFNet	0.4	0.7477	**0.8142**	0.7712	**0.7921**
		0.5	**0.7488**	0.7801	0.8007	0.7902
		0.6	0.7440	0.7436	0.8275	0.7833
		0.7	0.7308	0.6972	**0.8560**	0.7684
Corrosion	DeepLabV3+ (ResNet-101)	0.4	0.6758	**0.9559**	0.4777	0.6370
		0.5	0.6879	0.9499	0.4970	0.6525
		0.6	0.6986	0.9432	0.5155	0.6666
		0.7	**0.7093**	0.9350	**0.5352**	**0.6807**
	DeepLabV3+ (Xception)	0.4	0.6658	**0.9374**	0.4715	0.6274
		0.5	0.6872	0.9199	0.5057	0.6526
		0.6	0.7049	0.8983	0.5391	0.6738
		0.7	**0.7194**	0.8678	**0.5744**	**0.6912**
	DeepLabV3+ (Mobilenet)	0.4	0.6742	**0.9566**	0.4754	0.6351
		0.5	0.6850	0.9526	0.4922	0.6490
		0.6	0.6951	0.9479	0.5087	0.6620
		0.7	**0.7059**	0.9418	**0.5273**	**0.6760**
	U-Net	0.4	0.7038	**0.8825**	0.5610	0.6859
		0.5	0.7083	0.8662	0.5772	0.6927
		0.6	0.7111	0.8478	0.5930	0.6978
		0.7	**0.7122**	0.8226	**0.6113**	**0.7013**
	ERFNet	0.4	0.6742	**0.9303**	0.4847	0.6373
		0.5	0.6826	0.9241	0.4984	0.6475
		0.6	0.6886	0.9175	0.5123	0.6574
		0.7	**0.6973**	0.9088	**0.5284**	**0.6682**

[3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[5] M. Zhang, Y. Zhou, J. Zhao, Y. Man, B. Liu, and R. Yao, "A survey of semi-and weakly supervised semantic segmentation of images," *Artificial Intelligence Review*, pp. 1–30, 2019.

[6] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[7] Z. Huang, X. Wang, J. Wang, W. Liu, and J. Wang, "Weakly-supervised semantic segmentation network with deep seeded region growing," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7014–7023.

[8] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7268–7277.

[9] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation," in *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 275–12 284.

[10] M. Everingham and J. Winn, "The pascal visual object classes challenge 2012 (voc2012) development kit," *Pattern Analysis, Statistical Modelling and Computational Learning*, vol. 8, p. 5, 2011.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[12] A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 228–242, 2007.

[13] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *Proceedings of the 2016 European Conference on Computer Vision*, 2016, pp. 695–711.

[14] J. Ahn, S. Cho, and S. Kwak, "Weakly supervised learning of instance segmentation with inter-pixel relations," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*,

TABLE IV
PER-CLASS IOU ON VALIDATION SET OF PASCAL VOC 2012 SUBSET.

Dataset	Method	Background	Plane	Bike	Person	Car	Dog	Cat	Average (mIOU)
VOC subset	L_{pce}	0.7127	0.2427	0.0616	0.2871	0.1296	0.1626	0.3080	0,3121
	L_{cfm}	0.9084	0.6136	0.5901	0.6066	0.6420	0.7459	0.8082	0,7020
	$L_{\text{cfm}} + L_{\text{pce}}$	0.8651	0.5386	0.5296	0.6242	0.5884	0.6782	0.7395	0,6519
	L_{ce}	0.9379	0.7162	0.6439	0.6883	0.6882	0.7250	0.8552	0,7506

TABLE V
PER-CLASS IOU ON THE VALIDATION SET OF CITYSCAPE SUBSET.

Dataset	Method	Road	Building	Sidewalk	Traffic Sign	Plant	Sky	Person	Car	Bike	Average (mIOU)
CityScape subset	L_{pce}	0.7025	0.3666	0.5617	0.1207	0.6120	0.5599	0.1773	0.5513	0.2573	0.4343
	L_{cfm}	0.7561	0.5942	0.7083	0.4731	0.7623	0.7727	0.5723	0.6760	0.4503	0.6405
	$L_{\text{cfm}} + L_{\text{pce}}$	0.6763	0.5494	0.6816	0.4928	0.6418	0.6838	0.5774	0.5988	0.4024	0.5893
	L_{ce}	0.9084	0.7136	0.7301	0.6066	0.7420	0.7850	0.7082	0,6991	0.5296	0.7136

2019, pp. 2209–2218.

[15] T. Ma and A. Zhang, "Affinitynet: semi-supervised few-shot learning for disease type prediction," in *Proceedings of the 2019 AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 1069–1076.

[16] J. Choe and H. Shim, "Attention-based dropout layer for weakly supervised object localization," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2219–2228.

[17] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele, "Simple does it: Weakly supervised instance and semantic segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 876–885.

[18] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation," in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3136–3145.

[19] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," *Advances in Neural Information Processing Systems*, vol. 24, pp. 109–117, 2011.

[20] C.-C. Hsu, K.-J. Hsu, C.-C. Tsai, Y.-Y. Lin, and Y.-Y. Chuang, "Weakly supervised instance segmentation using the bounding box tightness prior," *Advances in Neural Information Processing Systems*, vol. 32, pp. 6586–6597, 2019.

[21] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Proceedings the 2001 IEEE international conference on computer vision*, vol. 1, 2001, pp. 105–112.

[22] L. Grady, "Random walks for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[23] X. Bai and G. Sapiro, "Geodesic matting: A framework for fast interactive image and video segmentation and matting," *International Journal of Computer Vision*, vol. 82, no. 2, pp. 113–132, 2009.

[24] M. Tang, F. Perazzi, A. Djelouah, I. Ben Ayed, C. Schroers, and Y. Boykov, "On regularized losses for weakly-supervised cnn segmentation," in *Proceedings of the 2018 European Conference on Computer Vision*, 2018, pp. 507–522.

[25] B. Wang, G. Qi, S. Tang, T. Zhang, Y. Wei, L. Li, and Y. Zhang, "Boundary perception guidance: a scribble-supervised semantic segmentation approach," in *Proceedings of the 2019 IJCAI International Joint Conference on Artificial Intelligence*, 2019.

[26] K. Yao, A. Ortiz, and F. Bonnin-Pascual, "A weakly-supervised semantic segmentation approach based on the centroid loss: Application to quality control and inspection," *IEEE Access*, vol. 9, pp. 69 010–69 026, 2021.

[27] O. Oktay, J. Schlemper, L. L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N. Y. Hammerla, B. Kainz, *et al.*, "Attention u-net: Learning where to look for the pancreas," in *Proceedings of the 2018 Medical Imaging with Deep Learning*, 2018.

[28] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the 2018 European conference on computer vision*, 2018, pp. 801–818.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proceedings of the 2015 International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015, pp. 234–241.

[30] E. Romera, J. M. Alvarez, L. M. Bergasa, and R. Arroyo, "Erfnet: Efficient residual factorized convnet for real-time semantic segmentation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 263–272, 2017.

[31] G. Papandreou, I. Kokkinos, and P.-A. Savalle, "Modeling local and global deformations in deep learning: Epitomic convolution, multiple instance learning, and sliding window detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 390–399.

[32] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251–1258.

[33] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 2169–2178.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[35] H. Qi, Z. Zhang, B. Xiao, H. Hu, B. Cheng, Y. Wei, and J. Dai, "Deformable convolutional networks–coco detection and segmentation challenge 2017 entry," in *Proceedings of the 2017 ICCV COCO Challenge Workshop*, vol. 15, 2017, p. 1.

[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the 2014 European Conference on Computer Vision*, 2014, pp. 740–755.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[38] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[39] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *Proceedings of the 2016 European Conference on Computer Vision*, 2016, pp. 525–542.

[40] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *arXiv preprint arXiv:1606.02147*, 2016.

[41] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015.

[42] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

VI. Supplementary

For clarity of exposition, a grey-scale example is used to illustrate the methodology of the closed-form matting algorithm [12]. In this section, the derivation of optimized target is shown in Section VII, and the formulation of the entry of the Matting Laplacian matrix is shown in Section VIII.

VII. The Optimized Target

For a gray-scale image, the value (I_i) of each pixel can be represented as

$$I_i = \alpha_i F_i + (1-\alpha_i)B_i \tag{24}$$

where F represents the foreground, B represents the background, and α denotes the alpha channel. Redoing (24), we can get

$$\alpha_i = \frac{1}{F_i - B_i} I_i + (-\frac{B_i}{F_i - B_i}) \tag{25}$$

Some assumptions on the nature of F, B and a are needed. Assume that both F and B are approximately constant over a small window around each pixel. This assumption allows us to rewrite (25), expressing α as a linear function of image I

$$\alpha_i \approx aI_i + b, \forall i \in w \tag{26}$$

where $a = \frac{1}{F-B}$, $b = -\frac{B}{F-B}$, and w is a small image window, whose size is 3×3 as usual. So the relation suggests finding α, a and b that minimizes the cost function

$$J(\alpha, a, b) = \sum_{j \in k} \left(\sum_{i \in w_k} (\alpha_i - a_j I_i - b_j)^2 + \epsilon a_j^2 \right) \tag{27}$$

where w_k is a small window around pixel j. The cost function includes a regularization term on a. One reason for this term is numerical stability [12].

The cost function can be written in matrix form as follows:

$$J(\alpha, a, b) = \sum_k \left|\left| \begin{bmatrix} I_1^j & 1 \\ I_2^j & 1 \\ \vdots & \vdots \\ I_w^j & 1 \\ \sqrt{\epsilon} & 0 \end{bmatrix} \begin{bmatrix} a_j \\ b_j \end{bmatrix} - \begin{bmatrix} \alpha_1^j \\ \alpha_2^j \\ \vdots \\ \alpha_w^j \\ 0 \end{bmatrix} \right|\right|^2 \quad j \in k \tag{28}$$

Let us define

$$G_k = \begin{bmatrix} I_1^j & 1 \\ I_2^j & 1 \\ \vdots & \vdots \\ I_w^j & 1 \\ \sqrt{\epsilon} & 0 \end{bmatrix}, \quad \overline{\alpha}_k = \begin{bmatrix} \alpha_1^j \\ \alpha_2^j \\ \vdots \\ \alpha_w^j \\ 0 \end{bmatrix} \quad j \in k \tag{29}$$

Then, the cost function changes to

$$J(\alpha, a, b) = \sum_k \left|\left| G_k \begin{bmatrix} a_k \\ b_k \end{bmatrix} - \overline{\alpha}_k \right|\right|^2 \tag{30}$$

For a given matte α, the optimal pair is

$$\begin{bmatrix} a_k^* \\ b_k^* \end{bmatrix} = \text{argmin} \left|\left| G_k \begin{bmatrix} a_k \\ b_k \end{bmatrix} - \overline{\alpha}_k \right|\right|^2 \tag{31}$$

Let $A = G_k$, $B = \overline{\alpha}_k$, $X = \begin{bmatrix} a_k \\ b_k \end{bmatrix}$, Hence, (31) changes to

$$X^* = \text{argmin} \, ||AX - B||^2 \tag{32}$$

where

$$\begin{aligned} ||AX-B||^2 &= (AX-B)^T(AX-B) \\ &= (X^TA^T - B^T)(AX-B) \\ &= X^TA^TAX - B^TAX - X^TA^TB + B^TB \\ &= X^TA^TAX - 2X^TA^TB + B^TB \end{aligned} \tag{33}$$

We compute the gradient of (33), and set it equal to 0 to obtain the optimal solution.

$$\begin{aligned} &\frac{\partial ||AX-B||^2}{\partial X} = 2A^TAX - 2A^TB = 0 \\ &\rightarrow A^TAX = A^TB \\ &\rightarrow X = (A^TA)^{-1}A^TB \end{aligned} \tag{34}$$

So, the optimal solution is

$$\begin{bmatrix} a_k^* \\ b_k^* \end{bmatrix} = (G_k^TG_k)^{-1}G_k^T\overline{\alpha}_k \tag{35}$$

Use the optimal solution $\begin{bmatrix} a_k^* \\ b_k^* \end{bmatrix}$ to replace (a, b) in the cost function $J(\alpha, a, b)$ in (30) as below:

$$\begin{aligned} J(\alpha) &= \sum_k \left|\left| G_k(G_k^TG_k)^{-1}G_k^T\overline{\alpha}_k - \overline{\alpha}_k \right|\right|^2 \\ &= \sum_k \left|\left| (I - G_k(G_k^TG_k)^{-1}G_k^T)\overline{\alpha}_k \right|\right|^2 \end{aligned} \tag{36}$$

Here, I is a identity matrix. Let $\overline{G}_k = I - G_k(G_k^TG_k)^{-1}G_k^T$, so $J(\alpha)$ can be written as:

$$\begin{aligned} J(\alpha) &= \sum_k ||\overline{G}_k\overline{\alpha}_k||^2 \\ &= \sum_k (\overline{G}_k\overline{\alpha}_k)^T\overline{G}_k\overline{\alpha}_k \\ &= \sum_k (\overline{\alpha}_k^T\overline{G}_k^T\overline{G}_k\overline{\alpha}_k) \end{aligned} \tag{37}$$

Let L represents the $\overline{G}_k^T\overline{G}_k$ and α refers to $\overline{\alpha}$, so $J(\alpha)$ is

$$J(\alpha) = \alpha^TL\alpha \tag{38}$$

The derivation of $L_{i,j}$ can be found in the next section.

So, the target is

$$\begin{aligned} &\min_\alpha J(\alpha) = \alpha^TL\alpha \\ &s.t. \quad (\alpha - S)^TD_c(\alpha - S) = 0 \end{aligned} \tag{39}$$

Here S represents the scribbles image containing the specified alpha values for the constrained pixels and zero for all other pixels, and the dimension of S is $N \times 1$. D_c is a diagonal matrix, which at the position of the scribble takes value 1 and for others taken value 0. The dimension of D_c is $N \times N$.

The Lagrange function $L(\alpha, \lambda)$ for this problem is

$$\begin{aligned} L(\alpha,\lambda) &= \alpha^T L\alpha + \lambda(\alpha^T - S^T)D_c(\alpha - S) \\ &= \alpha^T L\alpha + \lambda(\alpha^T D_c\alpha - S^T D_c\alpha \\ &\quad - \alpha^T D_c S + S^T D_c S) \\ \text{therefore} \quad \frac{\partial L(\alpha,\lambda)}{\partial \alpha} &= 2L\alpha + \lambda(2D_c\alpha - 2D_c S) \\ &= (L + \lambda D_c)2\alpha - 2\lambda D_c S \end{aligned} \tag{40}$$

Let the gradient be 0 to obtain the optimal solution,

$$\begin{aligned} &\frac{\partial L(\alpha,\lambda)}{\partial \alpha} = 0 \\ \rightarrow \quad &(L + \lambda D_c)2\alpha - 2\lambda D_c S = 0 \\ \rightarrow \quad &(L + \lambda D_c)\alpha - \lambda D_c S = 0 \\ \rightarrow \quad &(L + \lambda D_c)\alpha = \lambda D_c S \end{aligned} \tag{41}$$

Finally, the optimal solution α^* can be obtained by solving the following sparse linear system.

$$(L + \lambda D_c)\alpha - \lambda D_c S = 0 \tag{42}$$

where λ is some large number.

VIII. The Matting Laplacian Matrix

As known in previous section, L represents $\overline{G}_k^T \overline{G}_k$, where $\overline{G}_k = I - G_k(G_k^T G_k)^{-1} G_k^T$, here k represents the kth window, and I, α refers to the kth window.

$$G_k = \begin{bmatrix} I_1 & 1 \\ I_2 & 1 \\ \vdots & \vdots \\ I_w & 1 \\ \sqrt{\epsilon} & 0 \end{bmatrix} \qquad G_k^T = \begin{bmatrix} I_1 & I_2 & \cdots & I_w & \sqrt{\epsilon} \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix}$$

$$G_k^T G_k = \begin{bmatrix} I_1 & I_2 & \cdots & I_w & \sqrt{\epsilon} \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} I_1 & 1 \\ I_2 & 1 \\ \vdots & \vdots \\ I_w & 1 \\ \sqrt{\epsilon} & 0 \end{bmatrix}$$

$$G_k^T G_k = \begin{bmatrix} \sum_{i=0}^{w} I_i^2 + \epsilon & \sum_{i=0}^{w} I_i \\ \sum_{i=0}^{w} I_i & w \end{bmatrix} \tag{43}$$

As it is well known,

$$\begin{cases} \mu = \frac{1}{w}\sum_{i=0}^{w} I_i \\ \sigma^2 = \frac{1}{w}\sum_{i=0}^{w}(I_i - \mu)^2 \end{cases} \rightarrow \begin{cases} \sum_{i=0}^{w} I_i = \mu w \\ \sum_{i=0}^{w} I_i^2 = w\sigma^2 + w\mu^2 \end{cases} \tag{44}$$

So, (43) can be written as

$$G_k^T G_k = \begin{bmatrix} w\sigma^2 + w\mu^2 + \epsilon & w\mu \\ w\mu & w \end{bmatrix} \tag{45}$$

The inverse matrix $(G_k^T G_k)^{-1}$ is

$$\begin{aligned} &(G_k^T G_k)^{-1} \\ &= \frac{1}{w(w\sigma^2 + w\mu^2 + \epsilon) - \mu^2 w^2} \begin{bmatrix} w & -\mu w \\ -\mu w & w\sigma^2 + w\mu^2 + \epsilon \end{bmatrix} \\ &= \frac{1}{w^2\sigma^2 + w\epsilon} \begin{bmatrix} w & -\mu w \\ -\mu w & w\sigma^2 + w\mu^2 + \epsilon \end{bmatrix} \\ &= \frac{1}{w\sigma^2 + \epsilon} \begin{bmatrix} 1 & -\mu \\ -\mu & \sigma^2 + \mu^2 + \frac{\epsilon}{w} \end{bmatrix} \end{aligned} \tag{46}$$

Let $\frac{1}{w\sigma^2+\epsilon} = k_1$, $\sigma^2 + \mu^2 + \frac{\epsilon}{w} = k_2$, so $(G_k^T G_k)^{-1} = k_1 \begin{bmatrix} 1 & -\mu \\ -\mu & k_2 \end{bmatrix}$.

Therefore,

$$\begin{aligned} G_k(G_k^T G_k)^{-1} &= k_1 \begin{bmatrix} I_1 & 1 \\ I_2 & 1 \\ \vdots & \vdots \\ I_w & 1 \\ \sqrt{\epsilon} & 0 \end{bmatrix} \begin{bmatrix} 1 & -\mu \\ -\mu & k_2 \end{bmatrix} \\ &= k_1 \begin{bmatrix} I_1 - \mu & -I_1\mu + k_2 \\ I_2 - \mu & -I_2\mu + k_2 \\ \vdots & \vdots \\ I_w - \mu & -I_w\mu + k_2 \\ \sqrt{\epsilon} & -\mu\sqrt{\epsilon} \end{bmatrix} \end{aligned} \tag{47}$$

Therefore,

$$
\begin{aligned}
&G_k(G_k^TG_k)^{-1}G_k^T \\
&= k_1 \begin{bmatrix} I_1-\mu & -I_1\mu+k_2 \\ I_2-\mu & -I_2\mu+k_2 \\ \vdots & \vdots \\ I_w-\mu & -I_w\mu+k_2 \\ \sqrt{\epsilon} & -\mu\sqrt{\epsilon} \end{bmatrix} \begin{bmatrix} I_1 & I_2 & \cdots & I_w & \sqrt{\epsilon} \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \\
&= k_1 \begin{bmatrix} I_1I_1-\mu I_1-\mu I_1+k_2 & I_1I_2-\mu I_2-\mu I_1+k_2 & \cdots & I_1I_w-\mu I_w-\mu I_1+k_2 & \sqrt{\epsilon}I_1-\mu\sqrt{\epsilon} \\ I_2I_1-\mu I_1-\mu I_2+k_2 & I_2I_2-\mu I_2-\mu I_2+k_2 & \cdots & I_2I_w-\mu I_w-\mu I_2+k_2 & \sqrt{\epsilon}I_2-\mu\sqrt{\epsilon} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sqrt{\epsilon}I_1-\mu\sqrt{\epsilon} & \sqrt{\epsilon}I_2-\mu\sqrt{\epsilon} & \cdots & \sqrt{\epsilon}I_w-\mu\sqrt{\epsilon} & \epsilon \end{bmatrix}
\end{aligned} \tag{48}
$$

Since $\overline{G}_k = I - G_k(G_k^TG_k)^{-1}G_k^T$. Hence, the entry (i,j) of $\overline{G}_k(i,j)$ is

$$
\begin{aligned}
\overline{G}_k(i,j) &= \delta_{ij} - k_1(I_iI_j - \mu I_i - \mu I_j + k_2) \\
&= \delta_{ij} - (I_iI_j - \mu I_i - \mu I_j + \sigma^2 + \mu^2 + \frac{\epsilon}{w})\frac{1}{w\sigma^2+\epsilon} \\
&= \delta_{ij} - ((I_i-\mu)(I_j-\mu) + \sigma^2 + \frac{\epsilon}{w})\frac{1}{w\sigma^2+\epsilon} \\
&= \delta_{ij} - ((I_i-\mu)(I_j-\mu) + \frac{w\sigma^2+\epsilon}{w})\frac{1}{w\sigma^2+\epsilon} \\
&= \delta_{ij} - (\frac{1}{w\sigma^2+\epsilon}(I_i-\mu)(I_j-\mu) + \frac{1}{w}) \\
&= \delta_{ij} - \frac{1}{w}(1 + \frac{1}{\sigma^2+\frac{\epsilon}{w}}(I_i-\mu)(I_j-\mu))
\end{aligned} \tag{49}
$$

where δ_{ij} is the Kronecker delta,

$$
\delta_{ij} = \begin{cases} 0, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \tag{50}
$$

because $\overline{G}_k = I - G_k(G_k^TG_k)^{-1}G_k^T$

$$
\begin{aligned}
&\text{therefore } \overline{G}_k^T\overline{G}_k = (I - G_k(G_k^TG_k)^{-1}G_k^T)^T(I - G_k(G_k^TG_k)^{-1}G_k^T) \\
&= (I - G_k((G_k^TG_k)^{-1})^TG_k^T)(I - G_k(G_k^TG_k)^{-1}G_k^T) \\
&= I + G_k((G_k^TG_k)^{-1})^TG_k^TG_k(G_k^TG_k)^{-1}G_k^T - G_k((G_k^TG_k)^{-1})^TG_k^T - G_k(G_k^TG_k)^{-1}G_k^T \\
&= I + G_k((G_k^TG_k)^{-1})^TG_k^T - G_k((G_k^TG_k)^{-1})^TG_k^T - G_k(G_k^TG_k)^{-1}G_k^T \\
&= I - G_k(G_k^TG_k)^{-1}G_k^T \\
&= \overline{G}_k
\end{aligned} \tag{51}
$$

In the end, the $(i,j)th$ element in L matrix may be expressed as

$$
\delta_{ij} - \frac{1}{|w_k|}(1 + \frac{1}{\sigma^2+\frac{\epsilon}{|w_k|}}(I_i-\mu)(I_j-\mu)) \tag{52}
$$

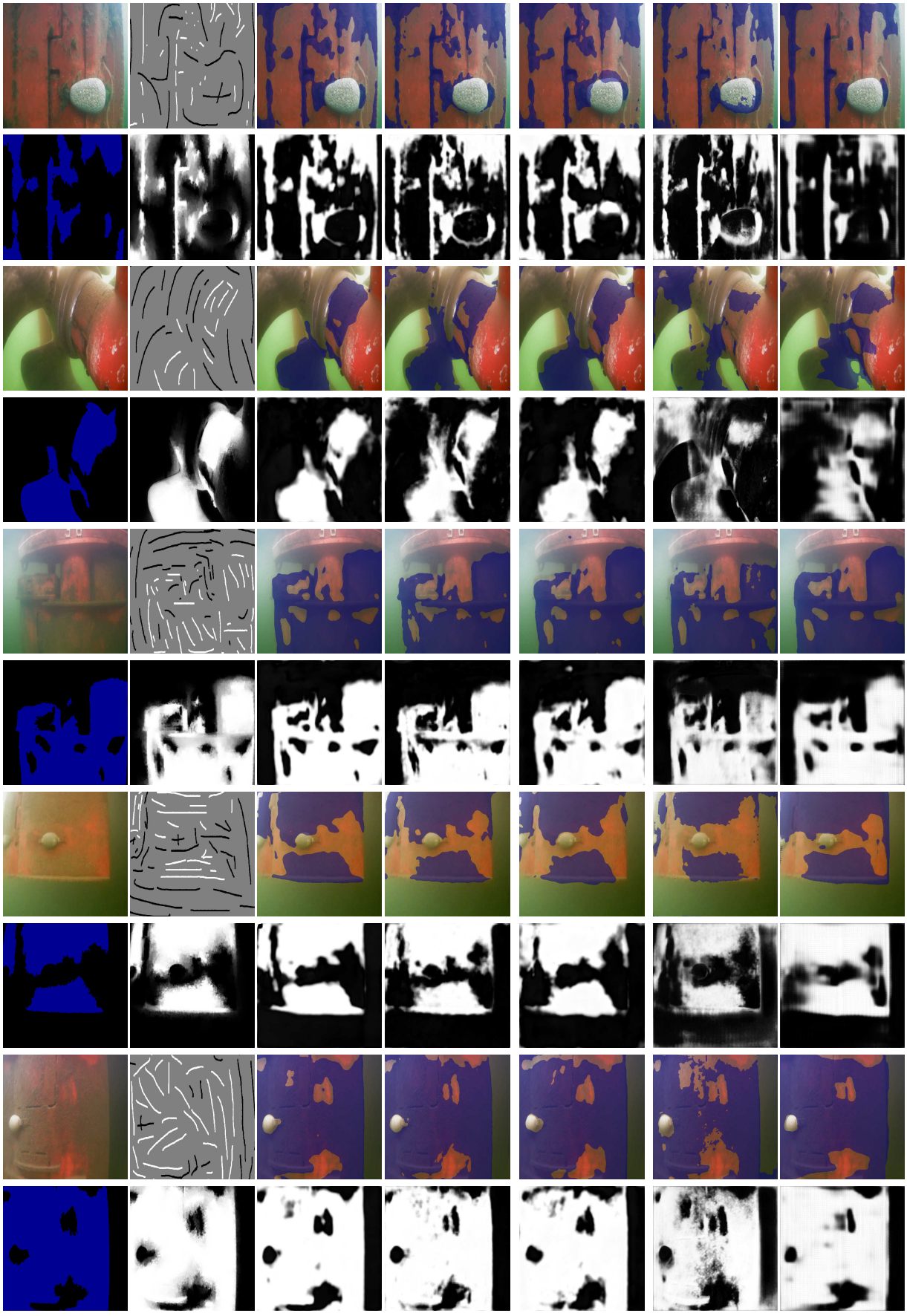


Fig. 8. Examples of segmentation and matting results for the BIO task. For every sample, the first image in the 1st row is the input image; and the 2nd image is the scribble annotation; from the 3- to 7-th, the images are the segmentation results using different segmentation network; in the second row, the first two images are the segmentation ground truth and matting results of the CFM algorithm; the rest images in the second row are the matting results of our approach by using different networks.

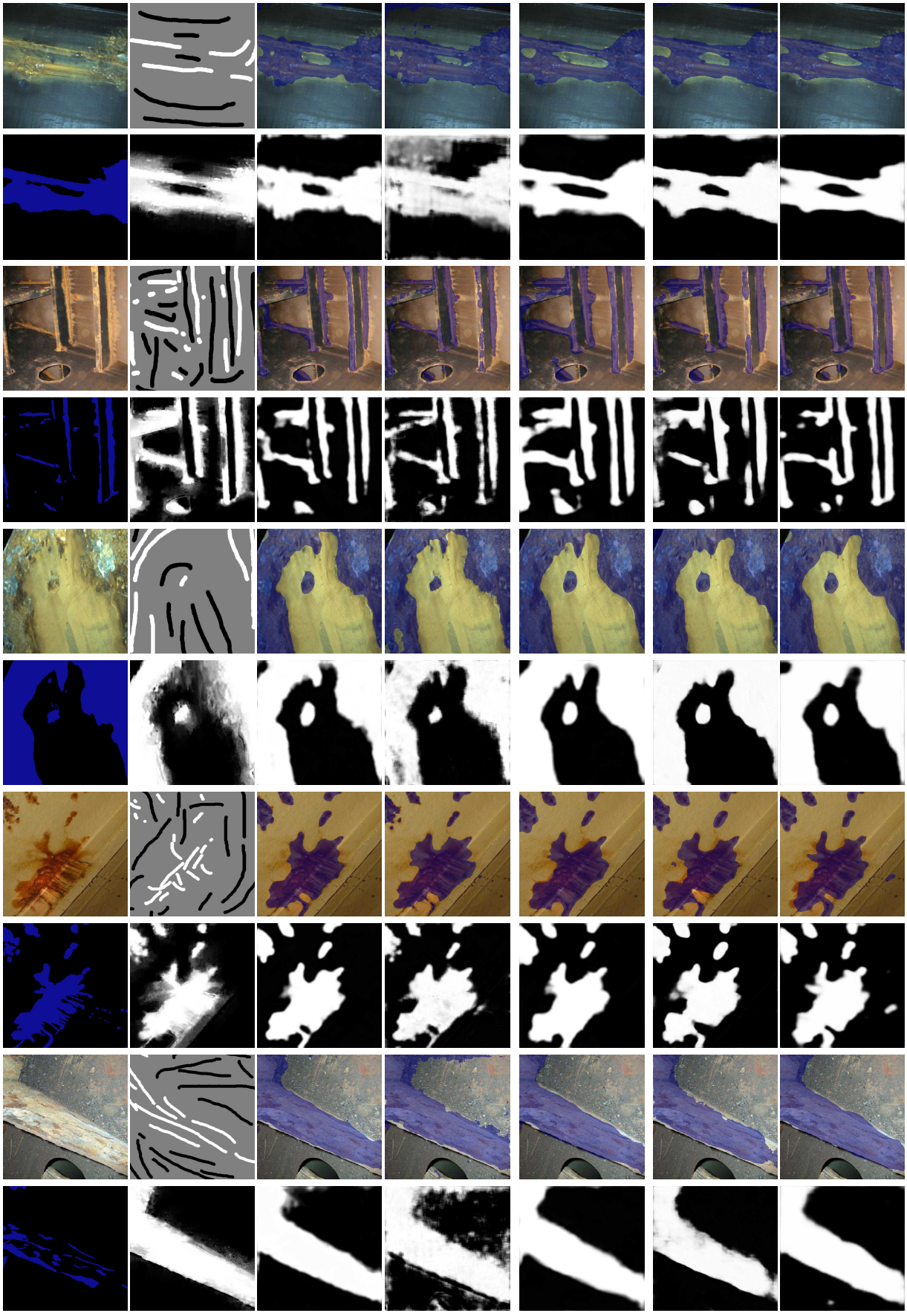


Fig. 9. Examples of segmentation and matting results for the COR task. For every sample, the first image in the 1st row is the input image; and the 2nd image is the scribble annotation; from the 3- to 7-th, the images are the segmentation results using different segmentation network; in the second row, the first two images are the segmentation ground truth and matting results of the CFM algorithm; the rest images in the second row are the matting results of our approach by using different networks.

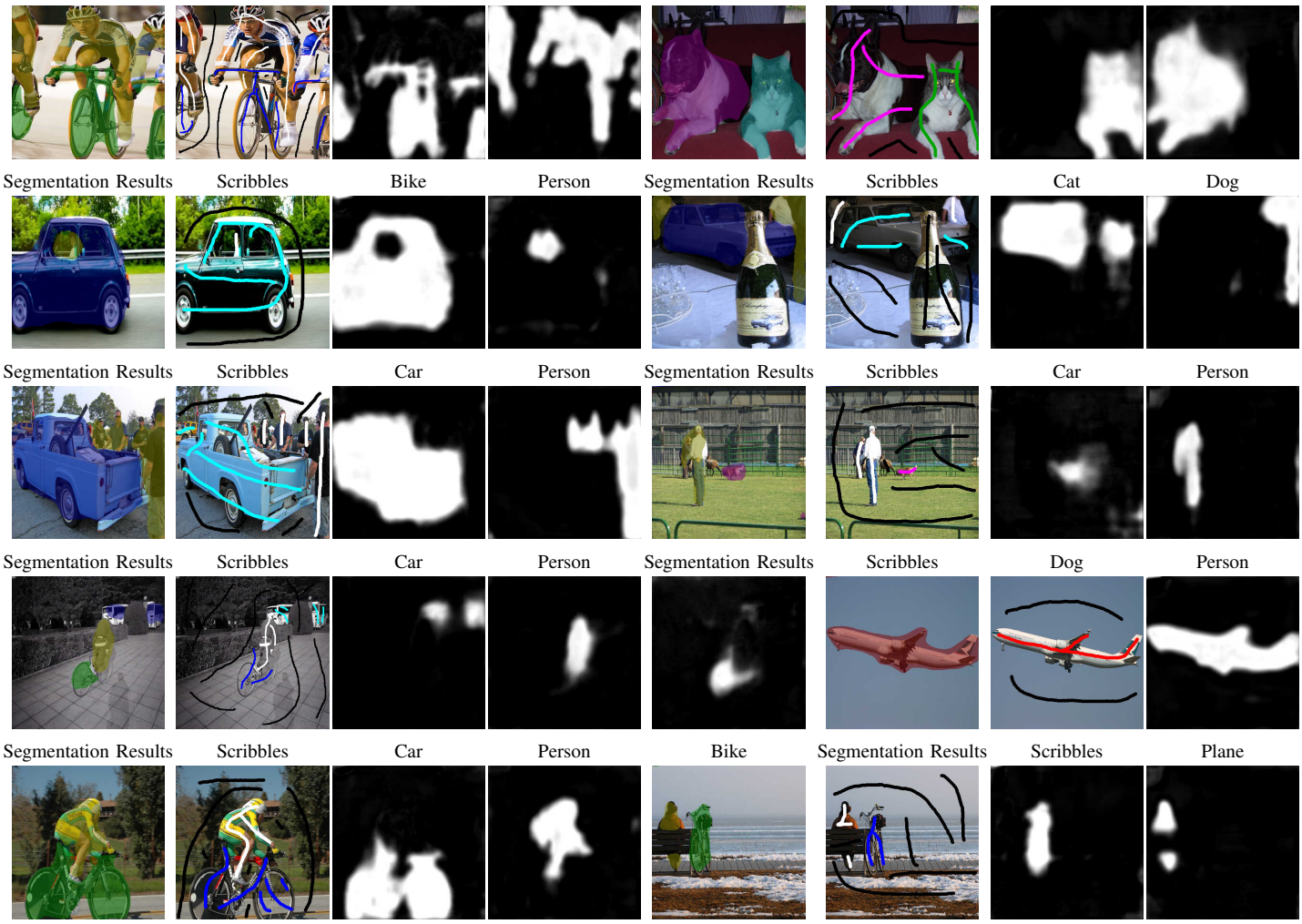


Fig. 10. Examples of segmentation and matting results for the VOC task. For every example, from left to right, the segmentation result is shown firstly; secondly, the scribble annotation is shown; the rest of images are the matting results of different categories.

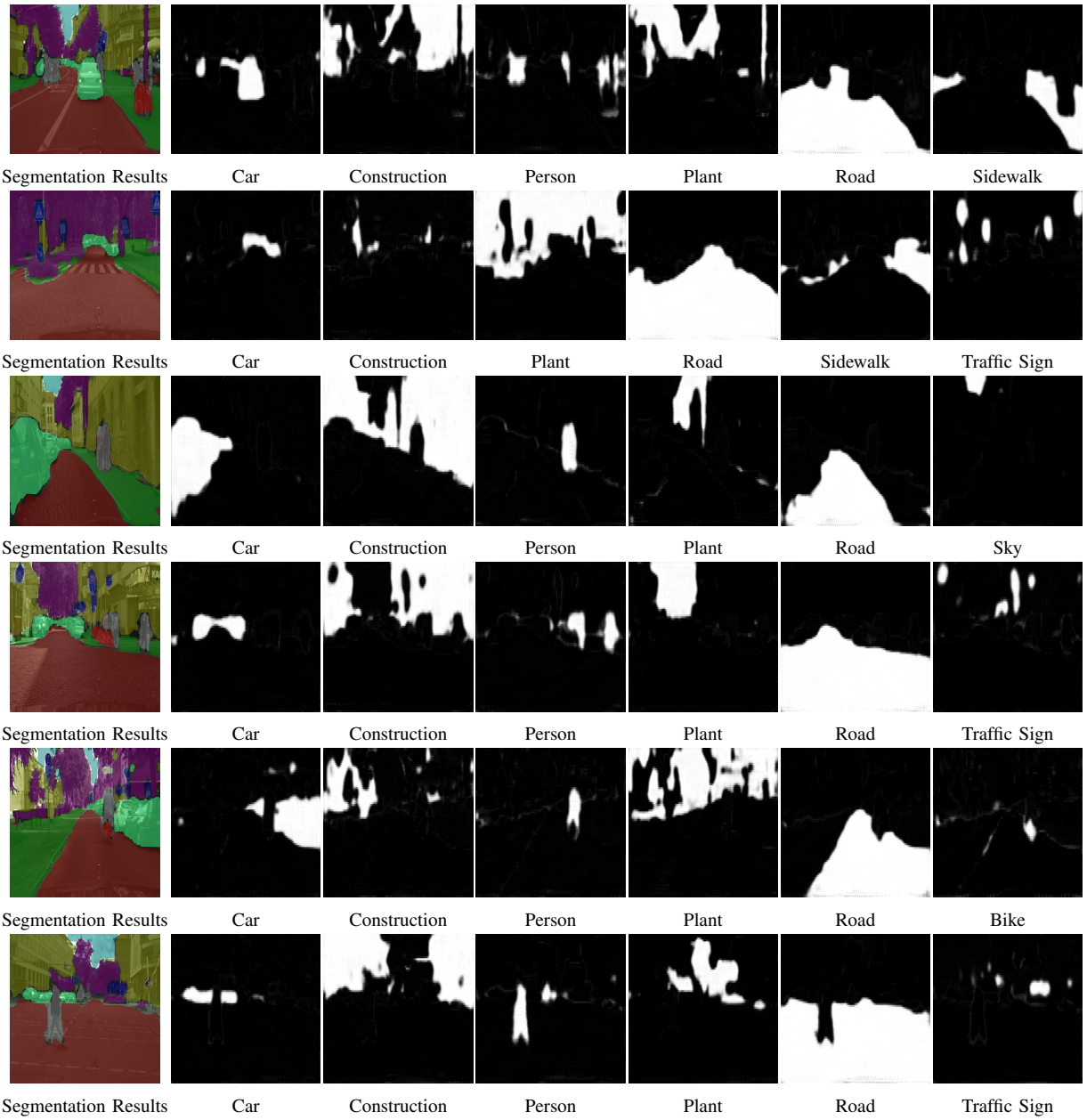


Fig. 11. Examples of segmentation and matting results for the CityScape task. For every example, from left to right, the segmentation result is shown firstly; then, the matting results of different categories are shown.