Article

Refined UNet V2: End-to-End Patch-Wise Network for Noise-Free Cloud and Shadow Segmentation

Libin Jiao, Lianzhi Huo, Changmiao Hu and Ping Tang *

Aerospace Information Research Institute (AIR), Chinese Academy of Sciences (CAS), Beijing 100101, China; jiaolb@aircas.ac.cn (L.J.); huolz@aircas.ac.cn (L.H.); hucm@aircas.ac.cn (C.H.)

* Correspondence: tangping@aircas.ac.cn

Received: 29 September 2020; Accepted: 22 October 2020; Published: 28 October 2020

Abstract: Cloud and shadow detection is an essential prerequisite for further remote sensing processing, whereas edge-precise segmentation remains a challenging issue. In Refined UNet, we considered the aforementioned task and proposed a two-stage pipeline to achieve the edge-precise segmentation. The isolated segmentation regions in Refined UNet, however, bring inferior visualization and should be sufficiently eliminated. Moreover, an end-to-end model is also expected to jointly predict and refine the segmentation results. In this paper, we propose the end-to-end Refined UNet v2 to achieve joint prediction and refinement of cloud and shadow segmentation, which is capable of visually neutralizing redundant segmentation pixels or regions. To this end, we inherit the pipeline of Refine UNet, revisit the bilateral message passing in the inference of conditional random field (CRF), and then develop a novel bilateral strategy derived from the Guided Gaussian filter. Derived from a local linear model of denoising, our v2 can considerably remove isolated segmentation pixels or regions, which is able to yield “cleaner” results. Compared to the high-dimensional Gaussian filter, the Guided Gaussian filter-based message-passing strategy is quite straightforward and easy to implement so that a brute-force implementation can be easily given in GPU frameworks, which is potentially efficient and facilitates embedding. Moreover, we prove that Guided Gaussian filter-based message passing is highly relevant to the Gaussian bilateral term in Dense CRF. Experiments and results demonstrate that our v2 is quantitatively comparable to Refined UNet, but can visually outperform that from the noise-free segmentation perspective. The comparison of time consumption also supports the potential efficiency of our v2.

Keywords: cloud and shadow segmentation; end-to-end UNet and CRF inference; Guided Gaussian filter

1. Introduction

More and more remote sensing applications are supported by cloud- and shadow-free images [1–4], while remote sensing images are usually degraded by clouds and cloud shadows, which leads to a negative effect on the further processing or resolve activity. In particular, cloud and shadow removal needs a necessary prerequisite of cloud and corresponding shadow segmentation, which is still a challenging issue in the remote sensing preprocessing. Fundamental solutions to the cloud and cloud shadow segmentation focus on manually developed segmentation methods, which can be generally grouped into three categories: spectral tests, temporal differentiation, and statistical methods [4]: spectral thresholds can be secured in terms of spectral data [3,5–9], temporal differentiation methods [10–12] pinpoint the movement of clouds and shadows, and statistical methods [13,14] exploit the statistics of spatial and spectral features. Manually developed methods can also be promoted by machine-learning methods due to large-scale labeled datasets [3], and data-driven segmentation methods have shown promising performance in cloud and shadow segmentation tasks [4,15].

Neural image segmentation (image segmentation by neural networks) approaches, on the other hand, introduce the learnable end-to-end solutions in the spatial and spectral feature spaces of remote sensing images. The convolutional neural network-based (CNN-based) feature encoders enable spatial and spectral feature extractions and output representative feature vectors, which can be used as the backbone for dense classification tasks. Learnable parameters [16–21] or network structures [22,23] push models to fit the feature space and reach the accurate pixel-wise classification results. Typical neural classifiers [4,24,25] have already been transferred to remote sensing segmentation. Some novel designing principles [23,26,27] can be gradually applied to segmentation models fitting particular scenarios as well.

Some challenging issues, however, remain in the cloud and cloud shadow segmentation tasks, such as the edge-precise segmentation [15]. Due to the discrete cost functions and the inflation of receptive fields in CNN-based feature extractors, neural image segmentation is restricted to coarse pixel-level classification, but the precise delineation of clouds and shadows is still limited. To achieve edge-precise segmentation, the fully connected conditional random field (Dense CRF) has been employed to model in pixel or patch level, and it is able to refine the segmentation performance on the edges. A feasible solution has been given in Refined UNet [15], in which we have preliminarily investigated the edge-precise cloud and shadow segmentation and proposed a feasible two-stage pipeline: the trainable UNet coarsely locates clouds and shadows patch by patch, and then the post-processing of Dense CRF refines the segmentation edges on the full images.

In the prospect of the two-stage Refined UNet [15], we expect an end-to-end implementation of UNet-CRF segmentation, which should incorporate UNet coarse prediction and CRF refinement for an individual patch in one forward step. However, the complicated high-dimensional filter-based bilateral message passing makes it difficult to extend Refined UNet to an end-to-end model, which inspires us to explore the bilateral message passing with other sophisticated strategies. Therefore, we intend to go deep into the Dense CRF and simplify its bilateral message passing, which is of help for building a joint implementation composed of UNet coarse prediction, unary transformation, and CRF refinement. In this paper, we inherit the two-stage pipeline of cloud and shadow segmentation in Refined UNet [15] and further explore an end-to-end solution, in which clouds and shadows can be jointly identified and refined by the concatenation of the pretrained UNet and the following CRF. Guided Gaussian filter-based message passing is employed in our computationally efficient CRF inference, rather than the complicated high-dimensional filter. Practically, a vanilla brute-force GPU implementation can be easily given by the Gaussian filter in GPU frameworks. Derived from the local linear model of denoising, our proposed CRF can effectively eliminate redundant isolated segmentation pixels or regions, which can yield “cleaner” results. A visual example of our Refined UNet v2 has been shown in Figure 1. Accordingly, our main contributions are listed as follows.

- Refined UNet V2: an experimental prototype of the end-to-end model for cloud and shadow segmentation is proposed, which can jointly predict and refine clouds and shadows by the concatenation of the UNet and following CRF for an individual image patch in one forward step.
- Straightforward and potentially efficient GPU implementation: we give an innovative Guided Gaussian filter-based message-passing strategy, which is straightforward and easy to implement in GPU frameworks. Thus, the vanilla implementation is also potentially efficient in computation.
- Noise-free segmentation: Our proposed Refined UNet v2 can effectively eliminate redundant isolated segmentation pixels or regions and yield “cleaner” results. Moreover, we demonstrate that the CRF can show a particular segmentation preference (edge-precise or clean results) if the bilateral term is customized to fit the preference.

Figure 1. Visualization of cloud and cloud shadow segmentation (Landsat 8 OLI, Path 113, Row 26, Year 2016), including false-color image, QA, Refined UNet [15], and our Refined UNet v2. Bands 5 NIR, 4 Red, and 3 Green are stacked as RGB channels to construct false-color images for visualization. Pixels of clouds and shadows with high confidence in the Level-2 Pixel Quality Assessment band are labeled as references (QA). Gray (#9C9C9C), green (#267300), and cyan (#73DFFF) mark pixels of background (0), cloud shadows (2), and clouds (3), respectively. Compared to Refined UNet [15], our Refined UNet v2 attempts to eliminate isolated pixels and regions, yielding a “cleaner” segmentation result.

The rest of the paper is organized as follows. Section 2 reviews related work regarding cloud and shadow segmentation. Proposed Refined UNet v2 is described in Section 3. Section 4 presents the experiments on Landsat 8 OLI dataset, including quantitative and visual comparisons against Refined UNet [15], ablation study with respect to the proposed CRF, hyperparameter sensitivity with respect to r and ϵ , and computational efficiency. Section 5 concludes this paper.

2. Related Work

In this section, we review semantic segmentation and related techniques, including neural image segmentation methods, CRF methods, and edge-preserving filters.

2.1. Neural Image Segmentation

In our paper, image segmentation by trainable neural networks is referred to as neural image segmentation, because the backbone of feature extraction and transformation is basically built upon convolutional [16–21] or graph neural networks [28,29]. Neural semantic segmentation methods mainly model the dense classification tasks by minimizing the pixel-level cost function [16,17]; they usually train an end-to-end segmentation model using pairs of images and labels. Neural semantic

segmentation methods generally benefit from CNN-based backbones, which effectively provide semantic feature transformation. Sophisticated CNN-based backbones, such as VGG-16/VGG-19 [30], MobileNets V1/V2/V3 [31–33], ResNet18/ResNet50/ResNet101 [34,35], and DenseNet [36], facilitate high-level semantic comprehension and then substantially promote the accuracy and efficacy of these tasks due to their delicate feature extraction.

Typical neural segmentation methods are reviewed as they can show the principles of designing architecture and adapting to particular scenarios. Fully convolutional network (FCN) [16] initiated neural semantic segmentation, which replaced fully connected layers with convolutional layers to adapt to the segmentation of arbitrary size. Unet [17] introduced the intermediate layer concatenation to reuse and fuse the extracted feature maps. Moreover, the long-range feature map aggregation was fully employed in segmentation tasks, such as RefineNet [18] enhancing high-resolution segmentation and PSPNet [19] with pyramid pooling module. In scene understanding applications, SegNet [20,21] inherited the encoder-decoder architecture for segmentation. For efficient computation in segmentation tasks, Joint pyramid upsampling was applied to FastFCN [37]. DeepLab series [38–41] exploited the atrous convolution, the CRF post-processing, the depth-wise separable convolution, the atrous spatial pyramid pooling module, and novel backbones, attempting to improve both the efficiency and robustness.

The concurrent trend of semantic segmentation concentrates on (i) bringing prior knowledge or features to particular scenarios, such as “cars cannot fly up in the sky” in urban scene segmentation [42], Fourier domain adaption [26], and model transfer (synthetic images to real images) [43], (ii) effective and efficient prediction, such as single-stage effective segmentation [44], boundary preserving segmentation [45], and very high-resolution segmentation [46], and (iii) novel network architecture, such as learnable dynamic architecture for semantic segmentation [23] and graph reasoning [27].

2.2. CRF-Based Image Segmentation

The CRFs can model semantic segmentation in pixel or patch level, which implicitly performs a maximum a posteriori (MAP) inference by minimizing the corresponding Gibbs energy function. Adjacency CRFs have been significantly improved by higher-order potentials or hierarchical connectivity, such as Robust P^n CRF [47,48], but they are still restricted to coarse segmentation due to the short-range connectivity. Fully connected CRFs benefit from the long-range connectivity, while the computation complexity remains a challenging problem; graph-cut [49] and the high-dimensional filter [50] are two common solutions to the efficient inference. On the other hand, the unary potentials in CRF can be predicted by a coarse segmentation structure. For example, CRFasRNN [51] presented an end-to-end segmentation structure concatenating CNN and CRF, in which the CNN-based segmentation backbone yielded the unary prediction and the following CRF refinement was built as a recurrent-neural-network (RNN) layer. And a similar combination of deep learning architecture and Gaussian Conditional Random Field (G-CRF) was given in [52]. CRFs can refine the segmentation performance in pixel level, which provides a novel perspective in our precise cloud and shadow segmentation task.

2.3. Edge-Preserving Filters

Edge-preserving filters aim to simultaneously smooth images and preserve edges, which have been widely used in applications of noise removal [53], high dynamic range (HDR) compression [54], haze removal [55], and joint upsampling [56]. Sophisticated filters include bilateral, weighted least squares (WLS), and guided filters. The bilateral filter [57] was a straightforward edge-preserving method to smooth images, which computed the output pixels weighted by a Gaussian kernel of both the spatial and color intensity discrepancy. The WLS [58] filter filtered images in an edge-preserving way of optimizing the quadratic function and taking as the guidance the input image, which achieved a global optimization. The guided filter [59] can perform the edge-preserving smoothness on images

by taking themselves as guidance. In our study, the edge-preserving filter provides a message-passing strategy of smoothing the segmentation and preserving the edges in the CRF inference.

3. Methodology

We introduce our Refined UNet v2 in four subsections, including an overview of Refined UNet v2 in Section 3.1, the Dense CRF as the segmentation refinement in Section 3.2, the Guided Gaussian filter as the efficient message-passing strategy in Section 3.3, and our end-to-end CRF inference in Section 3.4.

3.1. Overview of Refined UNet v2

We present the overview of our Refined UNet v2 performing noise-free segmentation on high-resolution remote sensing images. An end-to-end UNet-CRF architecture is used to roughly locate clouds and shadows and refine noise from a local perspective, which takes as input a seven-band 512×512 patch and yields a corresponding refined segmentation result. Hence, a high-resolution remote sensing image is first padded and cropped into patches of 512×512 , and then the aforementioned end-to-end network infers and refines patch by patch. The full segmentation result of clouds and shadows is eventually reconstructed from the patches. In this case, the pretrained UNet is inherited from [15] and the proposed CRF inference is introduced in the following subsections. The full pipeline of our Refined UNet v2 is illustrated in Figure 2.

Figure 2. The full pipeline of Refined UNet v2. First, patches are extracted from a high-resolution remote sensing image. UNet is employed to localize clouds and shadows roughly, and then the proposed CRF refines the prediction. The full result is restored from the refinement patches.

3.2. Revisiting Fully Connected Conditional Random Fields

We revisit the Dense CRF and the corresponding mean-field approximation inference, which has been thoroughly defined in [50]. Given the random field $\mathbf{X}$ and its global observation (image) $\mathbf{I}$, the CRF $(\mathbf{I}, \mathbf{X})$ is characterized by a Gibbs distribution, defined in Equation (1), and the corresponding Gibbs energy is given by Equation (2).

$$P(\mathbf{X}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp(-E(\mathbf{x}|\mathbf{I})) \quad (1)$$

$$E(\mathbf{x}) = \sum_i \psi_u(x_i) + \sum_i \sum_{i < j} \psi_p(x_i, x_j) \quad (2)$$

in which $\mathbf{x}$ denotes the label assignments for all pixels, ψ_u the unary potential, and ψ_p the pairwise potential.

In Equation (2), the unary potential can be practically given by a pixel-level classifier, and the pairwise potential is given by Equation (3).

$$\psi_p(x_i, x_j) = \mu(x_i, x_j) \underbrace{\sum_{m=1}^K w^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)}_{k(\mathbf{f}_i, \mathbf{f}_j)} \quad (3)$$

in which $k^{(m)}$ and $w^{(m)}$ denote a Gaussian kernel and its corresponding weight, $\mathbf{f}_i$ and $\mathbf{f}_j$ the feature vectors of pixel i and j , μ the label compatibility function.

In [50], the contrast-sensitive two-kernel potentials are given by Equation (4), in which I_i , I_j , p_i , and p_j denote color vectors and spatial positions of pixel i and j .

$$k(\mathbf{f}_i, \mathbf{f}_j) = w^{(1)} \exp \left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2} \right) + w^{(2)} \exp \left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2} \right) \quad (4)$$

The inference of CRF aims to find a $\hat{\mathbf{x}}$ as the most possible pixel-level classification by minimizing the energy function $E(\mathbf{x})$, and the mean-field approximation facilitates the inference instead of computing the exact distribution $P(\mathbf{X})$. Equation (5) of mean-field approximation leads to an iterative update algorithm, which is presented in Algorithm 1.

$$Q_i(x_i = l) = \frac{1}{Z_i} \exp \left\{ -\psi_u(x_i) - \sum_{l' \in \mathcal{L}} \mu(l, l') \sum_{m=1}^K w^{(m)} \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l') \right\} \quad (5)$$

Algorithm 1 Mean-Field Approximation in Fully Connected CRFs.

- 1: Initialize Q: $Q_i(x_i) \leftarrow \exp \{ -\psi_u(x_i) \} / Z_i$
 - 2: **while** not converged **do**
 - 3: $\tilde{Q}_i^{(m)}(l) \leftarrow \sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)$ for all m
 - 4: $\hat{Q}_i(x_i) \leftarrow \sum_{l \in \mathcal{L}} \mu^{(m)}(x_i, l) \sum_m w^{(m)} \tilde{Q}_i^{(m)}(l)$
 - 5: $Q_i(x_i) \leftarrow \exp \{ -\psi_u(x_i) - \hat{Q}_i(x_i) \}$
 - 6: normalize $Q_i(x_i)$
 - 7: **end while**
-

3.3. Guided Gaussian Filter

We introduce the Guided Gaussian filter as our proposed efficient message-passing method. As is presented in [50], the message passing is the bottleneck of efficient Dense CRF inference, and the high-dimensional filter based on the permutohedral lattice [60] is chosen to accelerate. The implementation of high-dimensional Gaussian filter significantly reduces the time complexity but is quite complicated. Therefore, we intend to find an alternative to efficient message passing which is highly relevant to the bilateral term, including both color intensity and Gaussian spatial feature. The Guided Gaussian filter can satisfy the requirement of the aforementioned bilateral term, which is introduced in this subsection.

Assuming a local linear model between the guidance I and the output y , we have Equation (6) in a window ω_k centered at the pixel k .

$$y_i = a_k I_i + b_k, \forall i \in \omega_k \quad (6)$$

The difference between the desired output y_i and the input x_i is assumed to the unwanted noise n_i , defined in Equation (7).

$$y_i = x_i - n_i \quad (7)$$

Given a guidance I , the solution should satisfy minimizing the discrepancy between the input x and the desired output y , which is formulated as a cost function of linear ridge regression. Moreover, we introduce a Gaussian weight g_{ik} in window ω_k , and the cost function is defined in Equation (9).

$$g_{ik} = \exp\left(-\frac{|p_i - p_k|^2}{\theta_\alpha^2}\right) \quad (8)$$

$$E(a_k, b_k) = \sum_{i \in \omega_k} g_{ik} \left((a_k I_i + b_k - x_i)^2 + \epsilon a_k^2 \right) \quad (9)$$

in which ϵ is a regularization parameter to penalize large a_k .

Equation (9) defines a linear ridge regression model and the solution is given by Equations (10)–(13).

$$a_k = \frac{\sum_{i \in \omega_k} g_{ik} I_i x_i - \sum_{i \in \omega_k} g_{ik} I_i \bar{x}'_k}{\sum_{i \in \omega_k} g_{ik} I_i^2 - \sum_{i \in \omega_k} g_{ik} I_i \mu'_k + \epsilon \sum_{i \in \omega_k} g_{ik}} \quad (10)$$

$$b_k = \frac{\sum_{i \in \omega_k} g_{ik} (x_i - a_k I_i)}{\sum_{i \in \omega_k} g_{ik}} = \bar{x}'_k - a_k \mu'_k \quad (11)$$

$$\bar{x}'_k = \frac{\sum_{i \in \omega_k} g_{ik} x_i}{\sum_{i \in \omega_k} g_{ik}} \quad (12)$$

$$\mu'_k = \frac{\sum_{i \in \omega_k} g_{ik} I_i}{\sum_{i \in \omega_k} g_{ik}} \quad (13)$$

in which $\bar{x}'_k$ and μ'_k denote the Gaussian-weighted means of x_i and I_i in window ω_k . The derivation is given in Appendix A.

In fact, a_k and b_k can be computed by the Gaussian filter with radius r , defined in Equations (14) and (15).

$$a_k = \frac{f_{\text{GF}}(I_i x_i) - f_{\text{GF}}(I_i) f_{\text{GF}}(x_i)}{f_{\text{GF}}(I_i^2) - f_{\text{GF}}^2(I_i) + \epsilon} \quad (14)$$

$$b_k = f_{\text{GF}}(x_i) - a_k f_{\text{GF}}(I_i) \quad (15)$$

in which the radius r controls the size of window ω_k : its width is $2r + 1$.

Using the Gaussian filter with radius r to compute the overlapping a_k and b_k , defined in Equations (16) and (17), y_i can be obtained by Equation (18) and the Guided Gaussian filter can be defined in Algorithm 2.

$$\bar{a}'_i = \frac{\sum_{k \in \omega_i} g_{ki} a_k}{\sum_{k \in \omega_i} g_{ki}} = f_{\text{GF}}(a_k) \quad (16)$$

$$\bar{b}'_i = \frac{\sum_{k \in \omega_i} g_{ki} b_k}{\sum_{k \in \omega_i} g_{ki}} = f_{\text{GF}}(b_k) \quad (17)$$

$$y_i = \bar{a}'_i I_i + \bar{b}'_i \quad (18)$$

Algorithm 2 Guided Gaussian Filter.

Input: Input image to filter x , Guidance image I , Radius r , Regularization parameter ϵ

Output: Filtered output y

```

1:  $\text{mean}'_I \leftarrow f_{\text{GF}}(I)$ 
2:  $\text{mean}'_x \leftarrow f_{\text{GF}}(x)$ 
3:  $\text{corr}'_I \leftarrow f_{\text{GF}}(I * I)$ 
4:  $\text{corr}'_{Ix} \leftarrow f_{\text{GF}}(I * x)$ 
5:  $\text{var}'_I \leftarrow \text{corr}'_I - \text{mean}'_I * \text{mean}'_I$ 
6:  $\text{cov}'_{Ix} \leftarrow \text{corr}'_{Ix} - \text{mean}'_I * \text{mean}'_x$ 
7:  $a \leftarrow \text{cov}'_{Ix} / (\text{var}'_I + \epsilon)$ 
8:  $b \leftarrow \text{mean}'_x - a * \text{mean}'_I$ 
9:  $\text{mean}'_a \leftarrow f_{\text{GF}}(a)$ 
10:  $\text{mean}'_b \leftarrow f_{\text{GF}}(b)$ 
11:  $y \leftarrow \text{mean}'_a * I + \text{mean}'_b$ 
12: return  $y$ 

```

where $*$ and $/$ denote element-wise multiplication and division.

A bilateral term composed of both the color intensity and the Gaussian spatial features is desired as that presented in [50]. Actually, the Guided Gaussian filter takes into consideration both the position and color intensity features of pixel i and j , which can be proved by the implicit kernel weights, defined in Equation (19).

$$W_{ij}(I) = \frac{1}{Z_g^2} \sum_{k: (i,j) \in \omega_k} g_{ik} g_{jk} \left(1 + \frac{(I_i - \mu'_k)(I_j - \mu'_k)}{\sigma'_k + \epsilon} \right) \quad (19)$$

in which Z_g , μ'_k , and σ'_k are the normalization term of the Gaussian filter $\sum_{i \in \omega_k} g_{ik}$, the Gaussian-weighted mean $f_{\text{GF}}(I_i)$ and variance $f_{\text{GF}}(I_i^2) - f_{\text{GF}}^2(I_i)$ in window ω_k , respectively. The proof is given in Appendix B.

3.4. End-to-End CRF Inference

As is mentioned above, an efficient inference has been presented in [50], in which a high-dimensional filtering algorithm was applied to the efficient bilateral message passing, defined in Equation (20). The high-dimensional filter of the permutohedral lattice [60] can reduce the time complexity of the bilateral message passing to linear time consumption ($O(N)$) [50].

$$\begin{aligned}
 \tilde{Q}_i^{(m)}(l) &= \underbrace{\sum_{j \neq i} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j) Q_j(l)}_{\text{message passing}} \\
 &= \underbrace{[G_{\Lambda^{(m)}} \otimes Q(l)](\mathbf{f}_i)}_{\tilde{Q}_i^{(m)}(l)} - Q_i(l)
 \end{aligned} \quad (20)$$

Nevertheless, the high-dimensional filter is quite complicated to implement, especially for GPU and end-to-end inference. Conversely, the brute-force Guided Gaussian filter can be applied to the

bilateral message passing, which is highly relevant to the bilateral features. The bilateral message passing term can be replaced by the Guided Gaussian filter while the smoothness term is still Gaussian kernel, defined in Equation (21).

$$\bar{Q}_i(l) = \underbrace{w^{(1)} \sum_{j \neq i} W_{ij}(l) Q_j(l)}_{\bar{Q}_i^{(1)}(l)} + \underbrace{w^{(2)} \sum_{j \neq i} k_{ij}^{(2)} Q_j(l)}_{\bar{Q}_i^{(2)}(l)} \quad (21)$$

in which the smoothness term is defined in Equation (22).

$$k_{ij}^{(2)} = \exp \left(-\frac{|p_i - p_j|^2}{2\theta_\gamma^2} \right) \quad (22)$$

Consequently, the mean-field approximation algorithm for our CRF inference is presented in Algorithm 3.

Algorithm 3 End-to-End Mean-Field Approximation in CRF Inference.

- 1: Initialize Q : $Q_i(x_i) \leftarrow \exp \{ -\phi_u(x_i) \} / Z_i$
 - 2: **while** not converged **do**
 - 3: $\bar{Q}_i^{(1)}(l) \leftarrow \sum_{j \neq i} W_{ij}(l) Q_j(l)$
 - 4: $\bar{Q}_i^{(2)}(l) \leftarrow \sum_{j \neq i} k_{ij}^{(2)} Q_j(l)$
 - 5: $\hat{Q}_i(x_i) \leftarrow \sum_{l \in \mathcal{L}} \mu^{(m)}(x_i, l) \sum_m w^{(m)} \bar{Q}_i^{(m)}(l)$
 - 6: $Q_i(x_i) \leftarrow \exp \{ -\psi_u(x_i) - \hat{Q}_i(x_i) \}$
 - 7: normalize $Q_i(x_i)$
 - 8: **end while**
-

4. Experiments and Discussion

We experiment with our method in several subsections, including quantitative and visual comparisons against Refined UNet [15], ablation study with respect to our CRF inference, hyperparameter sensitivity with respect to the r and ϵ in the CRF, and computational efficiency of the v2.

4.1. Revisiting Experimental Datasets, Preprocessing, Implementation Details, and Evaluation Metrics

To evaluate the performance of the proposed Refined UNet v2, we reuse the experimental dataset in [15], which is selected from Landsat 8 OLI imagery data [3]. Practically, test dataset, labels of clouds and shadows, class IDs, and visual settings are inherited: images of Landsat 8 OLI, Path 113, Row 26, Year 2016 are preserved for the test, listed as follows. For a fair comparison, the pretrained UNet backbone is inherited, which has been trained and validated on the training and validation sets given in [15]. QA is derived from the Level-2 Pixel Quality Assessment band as the reference, which specifically labels pixels of clouds and shadows with high confidence generated by the CFMask algorithm [8]. Please note that QA is referred to as reference rather than ground truth because the labels of clouds and shadows are dilated and not precise enough in the pixel level. Class IDs of background, fill values (invalid values, -9999), shadows, and clouds are assigned to 0, 1, 2, and 3, respectively; land, snow, and water are merged into background as we focus more on cloud and shadow segmentation. Band 5 NIR, 4 Red, and 3 Green are stacked as false-color images, and pixels of background (0), cloud shadows (2), and clouds (3) are colorized by gray (#9C9C9C), green (#267300), and cyan (#73DFFF), respectively. Please refer to [15] to obtain more details regarding the dataset.

- Test images of Landsat 8 OLI, Path 113, Row 26, Year 2016: 2016-03-27, 2016-04-12, 2016-04-28, 2016-05-14, 2016-05-30, 2016-06-15, 2016-07-17, 2016-08-02, 2016-08-18, 2016-10-21, and 2016-11-06

We also inherit the data preprocessing in [15]: fill and padded values are assigned to zero, full images are sliced into patches with a size of 512×512 , and all pixels in one patch are normalized to the interval $(0, 1]$ for UNet prediction. In our case, test images are padded to 8192×8192 so they can be sliced into 256 patches. UNet backbone takes all seven bands as input, but CRF only uses Bands 5, 4, and 3, and combines them to grayscale. We follow [50] to transform predictions into unary potentials.

In the implementation of CRF inference, the Gaussian and Guided Gaussian filters are built upon TensorFlow [61] framework, and the center weight of the Gaussian kernel is assigned to zero in order to not passing the message to itself. Instead of reusing the reported assessment, we reproduce Refined UNet [15] but the number of iteration of Dense CRF inference is set to 10. In our Refined UNet v2, ϵ is empirically assigned to 10^{-8} , while r varies from 10 to 80; we choose v2 of $r = 20$ for visual assessment as well. We also conduct a subsequent experiment to examine the effect with respect to these hyperparameters in this section. The demo code is available at <https://github.com/92xianshen/refined-unet-v2>.

For evaluation, we inherit the quantitative metrics from [15] to assess our methods, including accuracy, precision P , recall R , and F_1 scores. Considering the indicator p_{ij} denoting the cumulative number of pixels that should be grouped into class i but are actually grouped into class j , precision P reports the ability of the method to predict correctly, recall R reports the ability of the method to retrieve comprehensively, and F_1 score is a synthesized indicator considering both P and R . Equations (23)–(25) define above indicators.

$$P_i = \frac{p_{ii}}{\sum_{i=1}^C p_{ij}} \quad (23)$$

$$R_i = \frac{p_{ii}}{\sum_{j=1}^C p_{ij}} \quad (24)$$

$$F_{1i} = 2 \times \frac{P_i \cdot R_i}{P_i + R_i} \quad (25)$$

in which C is the number of all the classes.

Additionally, time consumption is also considered to be an indicator to evaluate these methods, which can demonstrate if our method is practically efficient.

4.2. Quantitative Comparison between Refined UNets and V2

We first clarify some expressions in the comparison experiments. For brevity, we refer to the segmentation results from UNet as predictions and those from CRF as refinements. The differences between local and global Refined UNet are refining on partial or full predictions. Hence, the local Refined UNet reproduces refinements by taking as input the image patches, and the full segmentation results are restored from patches. Conversely, the global Refined UNet predicts locally but refines globally, yielding full results by CRF. Naturally, the global Refined UNet represents the method in [15].

For quantitative comparison, we inherit the hyperparameter configuration of global and local Refined UNets from [15]: θ_α , θ_β , and θ_γ are empirically assigned to 80, 13, and 3, respectively. The quantitative assessment is listed in Table 1, in which precision, recall, and F_1 scores are compared. As can be seen in Table 1, the quantitative evaluations of UNet backbone, global Refined UNet [15], local Refined UNet, and Refined UNet v2 are approximately close in terms of accuracy, P , R , and F_1 . Please note that the QA band is in fact not precise enough as the ground truth, so we just take into account these quantitative indicators as the secondary evaluation, in order to observe if there is a distinct difference in numerical indicators. Accordingly, the quantitative assessment can demonstrate that our Refined UNet v2 is quantitatively comparable to Refined UNet [15]. On the other hand, the accuracy score drops slightly with the increase of the radius r , and the scores of precision P and

recall R of the cloud shadow decrease significantly. We attribute this to the intrinsic refinement of our v2: it prunes necessary shadows so that P and R are quite affected. Additionally, the side effect of these experiments is demonstrating the reproductivity of Refined UNet [15].

4.3. Visual Comparison between Refined UNet and V2

Furthermore, we compare the visual performance between Refined UNet [15] and v2. Theoretically, Refined UNet [15] focuses on global edge-preserving refinements while Refined UNet v2 concentrates more on eliminating noise in local predictions, due to their mathematical derivations. We present the qualitative evaluations as the principal assessment, which is illustrated in Figures 3 and 4. Visually, our Refined UNet v2 retains the prediction from its backbone but enlarges some small pieces of clouds and shadows from a global perspective (Figure 3). From local observation, it features “denoising” the prediction (eliminating isolated minor pieces or pixels), performing a better segmentation on the snow region (the first row in Figure 4). We attribute this to its intrinsic property of denoising, which has been shown in Section 3.3. Therefore, our Refined UNet v2 is superior in the noise-free segmentation of clouds and shadows.

Figure 3. Visualizations of cloud and cloud shadow segmentation (Landsat 8 OLI, Path 113, Row 26, Year 2016). (a) False-color image, (b) QA, (c) UNet backbone, (d) Refined UNet [15], (e) Refined UNet v2. Bands 5 NIR, 4 Red, and 3 Green are stacked as RGB channels to construct false-color images for visualization. Gray (#9C9C9C), green (#267300), and cyan (#73DFFF) mark pixels of background (0), cloud shadows (2), and clouds (3), respectively.

Table 1. Average accuracy, precision, recall, and F1 scores on the test set (Average $\pm$ Standard deviation, $^+$ represents that higher scores indicate better performance).

Models	Time (s/img) ¹	Acc. $^+$ (%)	Background (0)			Fill Values (1)			Shadows (2)			Clouds (3)		
			P $^+$ (%)	R $^+$ (%)	F1 $^+$ (%)	P $^+$ (%)	R $^+$ (%)	F1 $^+$ (%)	P $^+$ (%)	R $^+$ (%)	F1 $^+$ (%)	P $^+$ (%)	R $^+$ (%)	F1 $^+$ (%)
UNet [15,17] ²	20.67 $\pm$ 1.96	93.04 $\pm$ 5.45	93.34 $\pm$ 4.88	81.52 $\pm$ 15.3	86.35 $\pm$ 11.04	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	34.74 $\pm$ 14.77	54.31 $\pm$ 18.72	40.43 $\pm$ 14.74	87.28 $\pm$ 18.78	95.96 $\pm$ 3.63	90.12 $\pm$ 13.77
Global Refined UNet [15] ³	384.81 $\pm$ 5.91	93.48 $\pm$ 5.46	89.89 $\pm$ 7.39	85.94 $\pm$ 17.66	86.86 $\pm$ 12.33	99.88 $\pm$ 0.07	100 $\pm$ 0	99.94 $\pm$ 0.04	35.43 $\pm$ 20.26	17.87 $\pm$ 12.07	21.21 $\pm$ 11.89	87.6 $\pm$ 19.15	95.87 $\pm$ 3.2	90.15 $\pm$ 14.13
Local Refined UNet [15] ⁴	372.04 $\pm$ 5.01	93.64 $\pm$ 5.41	89.48 $\pm$ 7.48	87.04 $\pm$ 17.88	87.15 $\pm$ 12.28	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	40.80 $\pm$ 23.99	16.83 $\pm$ 10.40	21.24 $\pm$ 10.88	88.21 $\pm$ 19.02	94.92 $\pm$ 4.71	90.05 $\pm$ 13.90
Refined UNet v2 of $r = 10$	61.36 $\pm$ 5.25	93.60 $\pm$ 5.50	91.99 $\pm$ 5.74	84.51 $\pm$ 16.24	87.29 $\pm$ 11.35	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	40.64 $\pm$ 19.88	39.00 $\pm$ 13.77	36.79 $\pm$ 12.26	87.93 $\pm$ 18.83	95.83 $\pm$ 3.99	90.37 $\pm$ 13.89
Refined UNet v2 of $r = 20$	106.86 $\pm$ 2.47	93.56 $\pm$ 5.49	91.21 $\pm$ 6.36	85.18 $\pm$ 16.73	87.19 $\pm$ 11.59	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	39.55 $\pm$ 20.24	31.47 $\pm$ 11.77	32.12 $\pm$ 10.2	87.97 $\pm$ 18.88	95.65 $\pm$ 4.08	90.30 $\pm$ 13.91
Refined UNet v2 of $r = 30$	202.7 $\pm$ 2.7	93.51 $\pm$ 5.48	90.64 $\pm$ 6.85	85.62 $\pm$ 17.14	87.06 $\pm$ 11.78	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	37.94 $\pm$ 20.25	25.55 $\pm$ 11.03	27.55 $\pm$ 8.58	87.94 $\pm$ 18.91	95.52 $\pm$ 4.18	90.21 $\pm$ 13.92
Refined UNet v2 of $r = 40$	320.64 $\pm$ 4.77	93.47 $\pm$ 5.48	90.26 $\pm$ 7.16	85.92 $\pm$ 17.44	86.97 $\pm$ 11.93	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	36.24 $\pm$ 20.58	20.96 $\pm$ 10.68	23.54 $\pm$ 8.08	87.90 $\pm$ 18.93	95.41 $\pm$ 4.26	90.12 $\pm$ 13.94
Refined UNet v2 of $r = 50$	485.36 $\pm$ 2.91	93.45 $\pm$ 5.48	90.00 $\pm$ 7.35	86.13 $\pm$ 17.70	86.91 $\pm$ 12.09	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	35.07 $\pm$ 21.13	17.56 $\pm$ 10.21	20.34 $\pm$ 7.96	87.85 $\pm$ 18.96	95.31 $\pm$ 4.34	90.04 $\pm$ 13.96
Refined UNet v2 of $r = 60$	700.57 $\pm$ 17.64	93.42 $\pm$ 5.48	89.79 $\pm$ 7.48	86.28 $\pm$ 17.91	86.85 $\pm$ 12.22	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	34.01 $\pm$ 21.59	14.91 $\pm$ 09.47	17.71 $\pm$ 7.61	87.80 $\pm$ 18.97	95.21 $\pm$ 4.44	89.96 $\pm$ 13.97
Refined UNet v2 of $r = 70$	892.97 $\pm$ 5.36	93.40 $\pm$ 5.49	89.67 $\pm$ 7.55	86.37 $\pm$ 18.10	86.80 $\pm$ 12.34	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	33.13 $\pm$ 22.20	12.88 $\pm$ 08.75	15.60 $\pm$ 7.04	87.75 $\pm$ 19.00	95.13 $\pm$ 4.53	89.88 $\pm$ 13.98
Refined UNet v2 of $r = 80$	1213.23 $\pm$ 4.97	93.38 $\pm$ 5.49	89.57 $\pm$ 7.59	86.42 $\pm$ 18.25	86.75 $\pm$ 12.45	100 $\pm$ 0	100 $\pm$ 0	100 $\pm$ 0	32.22 $\pm$ 22.74	11.27 $\pm$ 08.16	13.85 $\pm$ 6.73	87.69 $\pm$ 19.01	95.06 $\pm$ 4.63	89.80 $\pm$ 13.99

¹ Inference duration for one full image in the test phase, s/img denotes seconds per image. ² Accuracy, precision, recall, and F1 scores of UNet [17] are reported in [15]. ³ The iteration of CRF in the reproduction of Refined UNet is 10, which leads to the minor differences of quantitative results, compared to [15]. ⁴ Local Refined UNet predicts patch by patch while global Refined UNet refines the entire image by Dense CRF.

However, it is noted that our Refined UNet v2 sometimes over-refines the “noise” of not only the misclassified minor regions (snow regions or fine cloud regions) but also the small pieces of background, which leads to merging two separated cloud regions; we attribute these drawbacks to the property of denoising as well. The strength of denoising can be essentially controlled by the radius r , which will be thoroughly explored in the following subsection. In addition, there remain some misclassified regions, a piece of the river, for example, is detected as the shadow, and some associated shadows are missing. This is, in our opinion, because of the misclassification of the UNet backbone and the strength of our CRF inference. In the future, we will further explore the property of other filter-based message-passing mechanisms to improve the CRF inference, and plan to introduce sophisticated weak-supervised strategies to improve the accuracy.

Figure 4. Local examples of cloud and shadow segmentation (Landsat 8 OLI, Path 113, Row 26, Year 2016, Date 12 April). (a) False-color patches, (b) QA, (c) UNet backbone, (d) Refined UNet [15], (e) Refined UNet v2. Bands 5 NIR, 4 Red, and 3 Green are stacked as RGB channels to construct a false-color image for visualization. Gray (#9C9C9C), green (#267300), and cyan (#73DFFF) mark pixels of background (0), cloud shadows (2), and clouds (3), respectively. Visually, Refined UNet [15] focuses more on the precise boundaries of clouds and shadows while generates more “noise” in the prediction. Refined UNet v2, alternatively, can remove the “noise”. Over-refinement, however, can also be found in terms of the visual assessment, which should be further considered in the future.

4.4. Ablation Study Regarding Our CRF Inference

We evaluate the performance of the UNet backbone and v2 to demonstrate the effect of our CRF inference, which is illustrated in Figures 3 and 4. Visually, UNet with adaptive weights concentrates more on the minority of categories (pixels of cloud shadows), which results in a prediction with too much noise. The visualizations in Figures 3 and 4 have been demonstrated above efficacy. Refined UNet [15] can refine the boundaries of cloud and shadow entities and, to some extent, eliminate isolated regions, but still preserve some stubborn fine regions. In contrast, v2 can faithfully

denoise these predictions and generate cleaner results. Consequently, our Refined UNet v2 outperforms Refined UNet [15] in the noise-free segmentation. Similarly, the over-refinement of isolated regions remains, which should be sometimes a drawback to overcome in future exploration.

4.5. Hyperparameter Sensitivity Regarding r and ϵ in Our CRF Inference

We evaluate the performance of Refined UNet v2 with respect to the hyperparameters r and ϵ . As is mentioned above, r and ϵ denote the radius of the Gaussian filter and the regularization parameter of the linear ridge regression model. According to [59], r controls the filter window and affects the inference efficiency of the brute-force implementation: a higher r should yield a more refined result because of longer-range connectivity but it takes more time in inference, while ϵ controls the ability to smooth predictions: a higher ϵ yields a “blur” prediction. The qualitative and quantitative assessments are illustrated and shown in Figure 5 and Table 1.

Figure 5. Visualizations with regards to r and ϵ in our CRF implementation. Gray (#9C9C9C), green (#267300), and cyan (#73DFFF) mark pixels of background (0), cloud shadows (2), and clouds (3), respectively. The candidate values of r vary from 10 to 80 when ϵ is secured to 10^{-8} . The candidate values of ϵ vary from 10^{-8} to 100 when r is secured to 50. The prediction is closer to boundaries if a higher r and a lower ϵ are used, while more isolated regions are also eliminated, especially for small pieces of shadows.

Figure 5 visually presents the effect of r in the performance of segmentation. As is speculated above, a higher r propagates messages with longer-range connectivity, leading to finer segmentation results; a lower r truncates long-range messages passing, leading to coarser segmentation results. The quantitative assessment in Table 1 also confirms the denoising sensitivity: v2 with a larger r prefers to prune more shadows, but it tends to retain more shadow entity if a smaller r is assigned to it. It can be visually explained by the similarity between shadows and segmentation noise. Our Refined UNet v2, on the other hand, is not sensitive with ϵ until it grows up to 1; the edges of segmentation results are not precise enough when ϵ is too high. In summary, we can increase r to yield cleaner results and decrease that to preserve more entities.

4.6. Computational Efficiency of Refined UNet v2

We compare the computational efficiency between Refined UNets and v2, which is shown in Table 1. Please note that the listed duration is the time consumption of inferring one full image in the test phase. As can be seen in Table 1, we can confirm that our v2 is potentially efficient compared to the global and local Refined UNets: it is noted that global and local Refined UNet spend similar time refining results and is not proportional to θ_α , θ_β , and θ_γ , but our v2 consumes less time than the counterparts if $r < 50$. The computational efficiency of our v2 benefits from the GPU support of TensorFlow framework. Moreover, according to Table 1, the computational efficiency of our brute-force implementation is highly related to the selection of radius r : the time consumption increases significantly when greater r is used, and we attribute this to our brute-force implementation. Thus, we will optimize the time performance by an efficient GPU filter in future work.

5. Conclusions

In this paper, we present an experimental prototype of an end-to-end pixel-level classifier for cloud and shadow noise-free patch-wise segmentation, which concatenates the UNet for cloud and shadow coarse segmentation and the following CRF inference for segmentation noise removal. In contrast to the separated pipeline of the local prediction of UNet and the global refinement of Dense CRF post-processing in [15], our end-to-end Refined UNet v2 locally gives the coarse prediction and the noise-free segmentation result simultaneously. Practically, it is straightforward for GPU frameworks and can be easily implemented by any machine-learning frameworks, which is able to be potentially efficient. Theoretically, we prove that the proposed bilateral term is highly relevant to both color intensity and Gaussian spatial features, which is similar to that of [50]. Experiments and results have demonstrated that our v2 is quantitatively comparable to Refined UNet [15] in terms of accuracy, precision, recall, and F1 scores, but can visually outperform that from the noise-free segmentation perspective. It is noted that a larger radius r results in the accuracy score dropping slightly, but the precision, recall, and F1 scores decreasing significantly, particularly for cloud shadows. We attribute this to the intrinsic refinement of our v2. The comparison of time consumption also supports that our implementation is potentially efficient; actually, our brute-force implementation is more efficient than Refined UNet if $r < 50$. We will further explore the backpropagation and the learnability of our CRF implementation, and improve its computational efficiency.

Author Contributions: Conceptualization, L.J. and P.T.; Funding acquisition, L.J.; Methodology, L.J.; Supervision, C.H. and P.T.; Writing—original draft, L.J.; Writing—review & editing, L.H. and C.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by China Postdoctoral Science Foundation grant number 2019M660852, Special Research Assistant Foundation of CAS, and National Natural Science Foundation of China grant number 41971396.

Acknowledgments: The authors would like to thank for the open-source implementations of the Cython-based Python wrapper for Dense CRF (<https://github.com/lucasb-eyer/pydensecrf>) and CRFasRNN-Keras (https://github.com/sadeepj/crfasrnn_keras) as they inspired the brute-force implementation of Refined UNet v2.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Derivation of Guided Gaussian Filter

We revisit the derivation of Guided Gaussian filter in this section and first recall the cost function of ridge regression with respect to linear coefficients a_k and b_k .

$$E(a_k, b_k) = \sum_{i \in \omega_k} g_{ik} \left((a_k I_i + b_k - x_i)^2 + \epsilon a_k^2 \right) \quad (\text{A1})$$

The partial derivatives of E with respect to a_k and b_k are given by Equations (A2) and (A3).

$$\frac{\partial E}{\partial a_k} = \sum_{i \in \omega_k} g_{ik} (2I_i(a_k I_i + b_k - x_i) + 2\epsilon a_k) \quad (\text{A2})$$

$$\frac{\partial E}{\partial b_k} = \sum_{i \in \omega_k} g_{ik} (2(a_k I_i + b_k - x_i)) \quad (\text{A3})$$

The two partial derivatives should be zero if we intend to minimize the cost function. Then we have

$$\frac{\partial E}{\partial a_k} = 0 \quad (\text{A4})$$

$$\frac{\partial E}{\partial b_k} = 0 \quad (\text{A5})$$

We first consider the partial derivative of E with respect to b_k and then we have

$$\sum_{i \in \omega_k} g_{ik} (2(a_k I_i + b_k - x_i)) = 0 \quad (\text{A6})$$

$$b_k \sum_{i \in \omega_k} g_{ik} = \sum_{i \in \omega_k} g_{ik} (x_i - a_k I_i) \quad (\text{A7})$$

Thus, b_k is given by Equation (A8). For brevity of exposition, we refer to μ'_k and $\bar{x}'_k$ as the Gaussian-weighted mean of I_i and x_i in the window ω_k .

$$\begin{aligned} b_k &= \frac{1}{\sum_{i \in \omega_k} g_{ik}} \sum_{i \in \omega_k} g_{ik} (x_i - a_k I_i) \\ &= \bar{x}'_k - a_k \mu'_k \end{aligned} \quad (\text{A8})$$

Now we consider the partial derivative of E with respect to a_k , given by Equations (A9) and (A10).

$$\sum_{i \in \omega_k} g_{ik} (2I_i(a_k I_i + b_k - x_i) + 2\epsilon a_k) = 0 \quad (\text{A9})$$

$$a_k \sum_{i \in \omega_k} g_{ik} I_i^2 + \sum_{i \in \omega_k} g_{ik} I_i b_k - \sum_{i \in \omega_k} g_{ik} I_i x_i + \epsilon a_k \sum_{i \in \omega_k} g_{ik} = 0 \quad (\text{A10})$$

We substitute b_k (Equation (A8)) into the term with respect to b_k in Equation (A10) and it yields

$$\begin{aligned} \sum_{i \in \omega_k} g_{ik} I_i b_k &= \sum_{i \in \omega_k} g_{ik} I_i (\bar{x}'_k - a_k \mu'_k) \\ &= \bar{x}'_k \sum_{i \in \omega_k} g_{ik} I_i - a_k \mu'_k \sum_{i \in \omega_k} g_{ik} I_i \end{aligned} \quad (\text{A11})$$

$$a_k \sum_{i \in \omega_k} g_{ik} I_i^2 + \bar{x}'_k \sum_{i \in \omega_k} g_{ik} I_i - a_k \mu'_k \sum_{i \in \omega_k} g_{ik} I_i - \sum_{i \in \omega_k} g_{ik} I_i x_i + \epsilon a_k \sum_{i \in \omega_k} g_{ik} = 0 \quad (\text{A12})$$

$$a_k \left(\sum_{i \in \omega_k} g_{ik} I_i^2 - \mu'_k \sum_{i \in \omega_k} g_{ik} I_i + \epsilon \sum_{i \in \omega_k} g_{ik} \right) = \sum_{i \in \omega_k} g_{ik} I_i x_i - \bar{x}'_k \sum_{i \in \omega_k} g_{ik} I_i \quad (\text{A13})$$

Now we have the solution to a_k , given by Equation (A14).

$$a_k = \frac{\sum_{i \in \omega_k} g_{ik} I_i x_i - \bar{x}'_k \sum_{i \in \omega_k} g_{ik} I_i}{\sum_{i \in \omega_k} g_{ik} I_i^2 - \mu'_k \sum_{i \in \omega_k} g_{ik} I_i + \epsilon \sum_{i \in \omega_k} g_{ik}} \quad (\text{A14})$$

Appendix B. Proof of the Kernel of Guided Gaussian Filter

The Guided Gaussian filter takes into consideration both the position and color intensity features of pixel i and j , which can be proved by the implicit kernel weights, defined in Equation (A15).

$$W_{ij}(I) = \frac{1}{Z_g^2} \sum_{k: (i,j) \in \omega_k} g_{ik} g_{jk} \left(1 + \frac{(I_i - \mu'_k)(I_j - \mu'_k)}{\sigma'_k + \epsilon} \right) \quad (\text{A15})$$

in which Z_g , μ'_k , and σ'_k are the normalization term of the Gaussian filter $\sum_{i \in \omega_k} g_{ik}$, the Gaussian-weighted mean $f_{\text{GF}}(I_i)$ and variance $f_{\text{GF}}(I_i^2) - f_{\text{GF}}^2(I_i)$ in window ω_k , respectively.

Proof. The kernel is given by Equation (A16).

$$W_{ij} = \frac{\partial y_i}{\partial x_j} \quad (\text{A16})$$

We replace b_k in Equation (18) with (15) and obtain

$$\begin{aligned} y_i &= f_{\text{GF}}(a_k) I_i + f_{\text{GF}}(b_k) \\ &= \frac{1}{Z_g} \sum_{k \in \omega_i} g_{ki} (a_k (I_i - \mu'_k) + \bar{x}'_k) \end{aligned} \quad (\text{A17})$$

Thus, the derivative of y_i with respect to x_j is

$$\frac{\partial y_i}{\partial x_j} = \frac{1}{Z_g} \sum_{k \in \omega_i} g_{ki} \left(\frac{\partial a_k}{\partial x_j} (I_i - \mu'_k) + \frac{\partial \bar{x}'_k}{\partial x_j} \right) \quad (\text{A18})$$

The derivative of $\bar{x}'_k$ with respect to x_j is

$$\frac{\partial \bar{x}'_k}{\partial x_j} = \frac{1}{Z_g} g_{jk} \delta_{j \in \omega_k} = \frac{1}{Z_g} g_{kj} \delta_{k \in \omega_j} \quad (\text{A19})$$

And the derivative of a_k with respect to x_j is

$$\begin{aligned} \frac{\partial a_k}{\partial x_j} &= \frac{1}{\sigma'_k + \epsilon} \left(\frac{\partial}{\partial x_j} f_{\text{GF}}(I_i x_i) - \mu'_k \frac{\partial \bar{x}'_k}{\partial x_j} \right) \\ &= \frac{1}{\sigma'_k + \epsilon} \left(\frac{1}{Z_g} g_{jk} I_j \delta_{j \in \omega_k} - \mu'_k \frac{1}{Z_g} g_{jk} \delta_{j \in \omega_k} \right) \\ &= \frac{1}{\sigma'_k + \epsilon} \left(\frac{1}{Z_g} g_{kj} (I_j - \mu'_k) \delta_{k \in \omega_j} \right) \end{aligned} \quad (\text{A20})$$

Finally, we have

$$\frac{\partial y_i}{\partial x_j} = \frac{1}{Z_g^2} \sum_{k \in \omega_i, k \in \omega_j} g_{ki} g_{kj} \left(1 + \frac{(I_i - \mu'_k)(I_j - \mu'_k)}{\sigma'_k + \epsilon} \right) \quad (\text{A21})$$

□

References

- Roy, D.P.; Wulder, M.A.; Loveland, T.R.; Woodcock, C.E.; Zhu, Z. Landsat-8: Science and product vision for terrestrial global change research. *Remote Sens. Environ.* **2014**, *145*, 154–172. [\[CrossRef\]](#)
- Wulder, M.A.; White, J.C.; Loveland, T.R.; Woodcock, C.E.; Roy, D.P. The global Landsat archive: Status, consolidation, and direction. *Remote Sens. Environ.* **2016**, *185*, 271–283. [\[CrossRef\]](#)
- Vermote, E.F.; Justice, C.O.; Claverie, M.; Franch, B. Preliminary analysis of the performance of the Landsat 8/OLI land surface reflectance product. *Remote Sens. Environ.* **2016**, *185*, 46–56. [\[CrossRef\]](#) [\[PubMed\]](#)
- Chai, D.; Newsam, S.; Zhang, H.K.; Qiu, Y.; Huang, J. Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks. *Remote Sens. Environ.* **2019**, *225*, 307–316. [\[CrossRef\]](#)
- Sun, L.; Liu, X.; Yang, Y.; Chen, T.; Wang, Q.; Zhou, X. A cloud shadow detection method combined with cloud height iteration and spectral analysis for Landsat 8 OLI data. *ISPRS J. Photogramm. Remote Sens.* **2018**, *138*, 193–207. [\[CrossRef\]](#)
- Qiu, S.; He, B.; Zhu, Z.; Liao, Z.; Quan, X. Improving Fmask cloud and cloud shadow detection in mountainous area for Landsats 4–8 images. *Remote Sens. Environ.* **2017**, *199*, 107–119. [\[CrossRef\]](#)
- Li, Z.; Shen, H.; Li, H.; Xia, G.; Gamba, P.; Zhang, L. Multi-feature combined cloud and cloud shadow detection in GaoFen-1 wide field of view imagery. *Remote Sens. Environ.* **2017**, *191*, 342–358. [\[CrossRef\]](#)
- Zhu, Z.; Woodcock, C.E. Object-based cloud and cloud shadow detection in Landsat imagery. *Remote Sens. Environ.* **2012**, *118*, 83–94. [\[CrossRef\]](#)
- Foga, S.; Scaramuzza, P.L.; Guo, S.; Zhu, Z.; Dilley, R.D.; Beckmann, T.; Schmidt, G.L.; Dwyer, J.L.; Hughes, M.J.; Laue, B. Cloud detection algorithm comparison and validation for operational Landsat data products. *Remote Sens. Environ.* **2017**, *194*, 379–390. [\[CrossRef\]](#)
- Zhu, X.; Helmer, E.H. An automatic method for screening clouds and cloud shadows in optical satellite image time series in cloudy regions. *Remote Sens. Environ.* **2018**, *214*, 135–153. [\[CrossRef\]](#)
- Frantz, D.; Roder, A.; Udelhoven, T.; Schmidt, M. Enhancing the Detectability of Clouds and Their Shadows in Multitemporal Dryland Landsat Imagery: Extending Fmask. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 1242–1246. [\[CrossRef\]](#)
- Zhu, Z.; Woodcock, C.E. Automated cloud, cloud shadow, and snow detection in multitemporal Landsat data: An algorithm designed specifically for monitoring land cover change. *Remote Sens. Environ.* **2014**, *152*, 217–234. [\[CrossRef\]](#)
- Ricciardelli, E.; Romano, F.; Cuomo, V. Physical and statistical approaches for cloud identification using Meteosat Second Generation-Spinning Enhanced Visible and Infrared Imager Data. *Remote Sens. Environ.* **2008**, *112*, 2741–2760. [\[CrossRef\]](#)
- Amato, U.; Antoniadis, A.; Cuomo, V.; Cuttillo, L.; Franzese, M.; Murino, L.; Serio, C. Statistical cloud detection from SEVIRI multispectral images. *Remote Sens. Environ.* **2008**, *112*, 750–766. [\[CrossRef\]](#)
- Jiao, L.; Huo, L.; Hu, C.; Tang, P. Refined UNet: UNet-Based Refinement Network for Cloud and Shadow Precise Segmentation. *Remote Sens.* **2020**, *12*, 2001. [\[CrossRef\]](#)
- Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *39*, 640–651.
- Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI, Munich, Germany, 5–9 October 2015*; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
- Lin, G.; Milan, A.; Shen, C.; Reid, I. RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017*.
- Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017*.

20. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [[CrossRef](#)] [[PubMed](#)]
21. Kendall, A.; Badrinarayanan, V.; Cipolla, R. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. In Proceedings of the British Machine Vision Conference, London, UK, 4–7 September 2017.
22. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning Transferable Architectures for Scalable Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
23. Li, Y.; Song, L.; Chen, Y.; Li, Z.; Zhang, X.; Wang, X.; Sun, J. Learning Dynamic Routing for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual Conference, Seattle, WA, USA, 14–19 June 2020.
24. Xie, F.; Shi, M.; Shi, Z.; Yin, J.; Zhao, D. Multilevel Cloud Detection in Remote Sensing Images Based on Deep Learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 3631–3640. [[CrossRef](#)]
25. Zi, Y.; Xie, F.; Jiang, Z. A Cloud Detection Method for Landsat 8 Images Based on PCANet. *Remote Sens.* **2018**, *10*, 877. [[CrossRef](#)]
26. Yang, Y.; Soatto, S. FDA: Fourier Domain Adaptation for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual Conference, Seattle, WA, USA, 14–19 June 2020.
27. Li, X.; Yang, Y.; Zhao, Q.; Shen, T.; Lin, Z.; Liu, H. Spatial Pyramid Based Graph Reasoning for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual Conference, Seattle, WA, USA, 14–19 June 2020.
28. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral Networks and Locally Connected Networks on Graphs. In Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada, 14–16 April 2014.
29. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems 29*; Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Roseville, CA, USA, 2016; pp. 3844–3852.
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018.
33. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
35. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. In *Computer Vision—ECCV 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 630–645.
36. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.
37. Wu, H.; Zhang, J.; Huang, K.; Liang, K.; Yu, Y. FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation. *arXiv* **2019**, arXiv:1903.11816.
38. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *arXiv* **2014**, arXiv:1412.7062.
39. Chen, L.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [[CrossRef](#)] [[PubMed](#)]

40. Chen, L.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. *arXiv* **2017**, arXiv:1706.05587.
41. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
42. Choi, S.; Kim, J.T.; Choo, J. Cars Can't Fly Up in the Sky: Improving Urban-Scene Segmentation via Height-Driven Attention Networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual Conference, Seattle, WA, USA, 14–19 June 2020.
43. Zhang, Y.; Qiu, Z.; Yao, T.; Ngo, C.W.; Liu, D.; Mei, T. Transferring and Regularizing Prediction for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual Conference, Seattle, WA, USA, 14–19 June 2020.
44. Araslanov, N.; Roth, S. Single-Stage Semantic Segmentation From Image Labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual Conference, Seattle, WA, USA, 14–19 June 2020.
45. Lee, H.J.; Kim, J.U.; Lee, S.; Kim, H.G.; Ro, Y.M. Structure Boundary Preserving Segmentation for Medical Image with Ambiguous Boundary. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual Conference, Seattle, WA, USA, 14–19 June 2020.
46. Cheng, H.K.; Chung, J.; Tai, Y.W.; Tang, C.K. CascadePSP: Toward Class-Agnostic and Very High-Resolution Segmentation via Global and Local Refinement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual Conference, Seattle, WA, USA, 14–19 June 2020.
47. Kohli, P.; Ladicky, L.; Torr, P.H.S. Robust Higher Order Potentials for Enforcing Label Consistency. *Int. J. Comput. Vis.* **2009**, *82*, 302–324. [[CrossRef](#)]
48. Ladicky, L.; Russell, C.; Kohli, P.; Torr, P.H. Associative Hierarchical CRFs for Object Class Image Segmentation. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 29 September–2 October 2009.
49. Kolmogorov, V.; Zabini, R. What energy functions can be minimized via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 147–159. [[CrossRef](#)]
50. Krähenbühl, P.; Koltun, V. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. In *Advances in Neural Information Processing Systems 24*; Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Roseville, CA, USA, 2011; pp. 109–117.
51. Zheng, S.; Jayasumana, S.; Romeraparedes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H.S. Conditional Random Fields as Recurrent Neural Networks. In Proceedings of the International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1529–1537.
52. Chandra, S.; Kokkinos, I. Fast, Exact and Multi-scale Inference for Semantic Image Segmentation with Deep Gaussian CRFs. In *Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 8–16 October 2016*; Leibe, B., Matas, J., Sebe, N., Welling, M., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 402–418.
53. Liu, C.; Freeman, W.T.; Szeliski, R.; Kang, S.B. Noise Estimation from a Single Image. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006.
54. Durand, F.; Dorsey, J. Fast Bilateral Filtering for the Display of High-Dynamic-Range Images. *ACM Trans. Graph.* **2002**, *21*, 257–266. [[CrossRef](#)]
55. He, K.; Sun, J.; Tang, X. Single Image Haze Removal Using Dark Channel Prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2341–2353.
56. Kopf, J.; Cohen, M.F.; Lischinski, D.; Uyttendaele, M. Joint bilateral upsampling. *ACM Trans. Graph.* **2007**, *26*, 96. [[CrossRef](#)]
57. Tomasi, C.; Manduchi, R. Bilateral filtering for gray and color images. In Proceedings of the Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), Bombay, India, 4–7 January 1998; pp. 839–846.
58. Farbman, Z.; Fattal, R.; Lischinski, D.; Szeliski, R. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph.* **2008**, *27*. [[CrossRef](#)]
59. He, K.; Sun, J.; Tang, X. Guided Image Filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1397–1409. [[CrossRef](#)] [[PubMed](#)]

60. Adams, A.; Baek, J.; Davis, M.A. Fast High-Dimensional Filtering Using the Permutohedral Lattice. *Comput. Graph. Forum* **2010**, *29*, 753–762. [[CrossRef](#)]
61. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software. 2015. Available online: tensorflow.org (accessed on 27 October 2020).

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).