applied sciences

MDPI

Article

Enhanced Atrous Extractor and Self-Dynamic Gate Network for Superpixel Segmentation

Bing Liu [1,2], Zhaohao Zhong [2], Tongye Hu [2] and Hongwei Zhao [1,*]

[1] College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2] School of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China; 2202103113@stu.ccut.edu.cn (Z.Z.)
* Correspondence: zhaohw@jlu.edu.cn

Abstract: A superpixel is a group of pixels with similar low-level and mid-level properties, which can be seen as a basic unit in the pre-processing of remote sensing images. Therefore, superpixel segmentation can reduce the computation cost largely. However, all the deep-learning-based methods still suffer from the under-segmentation and low compactness problem of remote sensing images. To fix the problem, we propose EAGNet, an enhanced atrous extractor and self-dynamic gate network. The enhanced atrous extractor is used to extract the multi-scale superpixel feature with contextual information. The multi-scale superpixel feature with contextual information can solve the low compactness effectively. The self-dynamic gate network introduces the gating and dynamic mechanisms to inject detailed information, which solves the under-segmentation effectively. Massive experiments have shown that our EAGNet can achieve the state-of-the-art performance between k-means and deep-learning-based methods. Our methods achieved 97.61 in ASA and 18.85 in CO on the BSDS500. Furthermore, we also conduct the experiment on the remote sensing dataset to show the generalization of our EAGNet in remote sensing fields.

Keywords: superpixel segmentation; gating mechanism; multi-scale superpixel feature

Citation: Liu, B.; Zhong, Z.; Hu, T.; Zhao, H. Enhanced Atrous Extractor and Self-Dynamic Gate Network for Superpixel Segmentation. *Appl. Sci.* **2023**, *13*, 13109. https://doi.org/10.3390/app132413109

Academic Editors: Francesco Zirilli and Andrea Prati

Received: 13 October 2023
Revised: 24 November 2023
Accepted: 1 December 2023
Published: 8 December 2023

1. Introduction

A superpixel is a group of pixels with similar color, texture, and low-level and mid-level properties. Superpixel segmentation aims to divide the image with several superpixels, which can reduce the basic primitive effectively. Therefore, the computation cost could be reduced vastly.

Recently, the superpixel segmentation algorithm has been applied in remote sensing fields, which is used to reduce the dimension of features to speed up the training and inference time. Therefore, superpixel segmentation can open up some new scenarios in remote sensing. An example of this is ESCNet [1], which introduces a superpixel to reduce the latent noise of the pixel-level feature maps while preserving the edges. SG-waterNet [2] introduces superpixels to produce a superpixel graph, which contains more powerful context information that can be exploited by the GCN. The MAST [3] model takes advantage of the adaptive spatial nature of superpixels to achieve better classification performance with high-resolution remotely sensed images. With these applications of superpixel segmentation algorithms [4–10], superpixel segmentation has become a key technology in the remote sensing of the computer vision field.

However, all these applications of superpixel methods introduce the traditional k-means superpixel segmentation algorithm [11–14], which still suffers from the hand-craft feature and is non-differentiable. An example of this is SLIC [15], which first initializes the seed and computes the associate map between the seeds and the surrounding pixels. SNIC [16] introduces a priority queue to assign the pixels to the correct seeds. LSC [17] introduces mapping the property to the high-dimension space to obtain the superpixels. All

these traditional k-means superpixel segmentation algorithms are difficult to incorporate into the convolutional neural network and can not get the accuracy of the superpixel map.

In order to fix the problems, some deep-learning-based methods are proposed to fix the superpixel segmentation. An example of this is SSN [18], which computes a differentiable soft associate map between pixels and seeds. SCN [19] first proposes an end-to-end superpixel segmentation network. However, all these deep-learning-based methods still suffer from under-segmentation and low compactness in remote sensing images.

To solve these problems, we propose EAGNet, the enhanced atrous extractor and self-dynamic gate network. The enhanced atrous extractor introduces our proposed enhanced atrous convolution and transformer architecture based on a multi-scale pixel feature to extract a multi-scale superpixel feature with contextual information. In particular, the enhanced atrous extractor first introduces the atrous convolution with the SiLU function to extract the multi-scale superpixel feature and feed it into the MLP. The self-dynamic gate network introduces the gating and dynamic mechanism to inject the pixel information. Specially, the self-dynamic gate network introduces the convolution and sigmoid to produce the gate of the pixel and the superpixel feature by themselves. The multi-scale superpixel feature with contextual information is useful for solving the low compactness problem. Our self-dynamic gate can solve the under-segmentation of remote sensing images. We conduct massive experiments on the BSDS500 dataset [20] and UCM dataset [21] to show that our EAGNet can not only fix the under-segmentation and low compactness of remote sensing images but can also achieve state-of-the-art performance among traditional k-means superpixel segmentation and deep-learning-based algorithms. We also conduct numerous ablation studies to prove the effectiveness of our proposed method.

Our main contributions can be listed as follows:

(1) We propose an enhanced atrous extractor, which introduces enhanced atrous convolution based on a transformer architecture to extract multi-scale superpixel features with contextual information.
(2) We propose a self-dynamic gate network, which introduces a gating and dynamic mechanism to inject detailed information.
(3) Our EAGNet can achieve the state-of-the-art performance among traditional k-means superpixel segmentation and deep-learning-based algorithms.

2. Related Work

Superpixel segmentation: Superpixel segmentation aims to group pixels with similar low- and mid-level properties. We consider the group of pixels as a superpixel, which can reduce the computation cost. Traditional superpixel segmentation algorithms mainly introduce k-means-based methods, which compute the associate map between the seed and surrounding pixels. SLIC [15] initializes the seeds and computes the associate map of the pixels and the superpixel. Then, it assigns every pixel a label based on the associate map. Finally, it computes the average of the pixels labeled to define a new seed. LSC [17] first maps the RGB image to the 10-dimension feature space and computes the associate map. SNIC [16] first initializes centroids and uses a priority queue to assign the pixels to the correct centroid. The manifold SLIC [15] introduces a two-dimension manifold to compute a content-sensitive superpixel map. However, these k-means-based methods are non-differentiable and can not be incorporated into the convolutional neural network. Therefore, to fix the problem, some deep-learning methods are proposed. SSN [18] proposes a differentiable soft associate map and introduces a convolutional neural network to extract features. The SCN [19] first proposes an end-to-end Unet architecture to predict the superpixel map. However, these methods often result in the under-segmentation and low compactness of the remote sensing image. To fix the problem, we propose EAGNet, an enhanced atrous extractor and self-dynamic gate network. The enhanced atrous extractor can extract the multi-scale superpixel feature, and the self-dynamic gate can fuse the feature dynamic, which can fix the low compactness and under-segmentation, respectively.

Vision transformer: Superpixel segmentation aims to group pixels with similar low- and mid-level properties. We consider the group of pixels as a superpixel, which can reduce the computation cost. A traditional transformer is an effective technology that was first proposed by the natural language process field. The development of the transformer raised the attention of the computer vision field rapidly. Vit is the first vision transformer model in the computer vision field. Vit [22] introduces the convolution to divide the 16×16 dimension of the patch into an embedding space and compute the self-attention of these patches. Swin transformer [23] proposes the swift window self-attention and hierarchy transformer to learn the powerful feature representation. PVT [24] first combines convolution and self-attention to reduce the dimension of the feature and provide the multi-scale feature for the downstream task. However, all these methods still suffer from the huge computation cost. In order to reduce the computation cost, some lightweight methods are proposed to reduce the computation cost. An example of this is MiniVit [25], which introduces the distillation and teacher–student model to achieve weight multiplexing, which reduces the computation cost largely. The Davit [26] introduces spatial-wise self-attention and channel-wise self-attention to reduce the computation cost. EfficientNet [27] combines the CNN block and transformer block to reduce the computed number of self-attention. However, all these methods still suffer from high latency, and it is hard to extract the multi-scale feature efficiently. To fix these problems, we propose the enhanced atrous extractor, which is a transformer-based architecture but a pure CNN feature extractor. Different from previous methods, we introduce the non-local atrous convolution to replace the self-attention to extract the multi-scale superpixel feature with context information. The latency of our proposed enhanced atrous extractor can satisfy the requirement of superpixel segmentation.

Gating mechanism: The gating mechanism is a technology that can control the passing of information. It was first proposed by LSTM [28], which is the basic block of RNN [29]. In order to reduce the computation complexity, they propose a GRU [30] to control the passing of information. Recently, some works have introduced the gating mechanism to filter the feature, such as GateNet [31], which introduces feature embedding and hidden gates to obtain the high-order interaction information. DepthNet introduces the gating mechanism to adjust the dimension of the feature adaptively. GFF [32] uses the gating mechanism to select multi-scale features. GSCNet [33] connects the two-branch information by using a gating mechanism. However, these methods can only filter the feature or not fuse the feature. Therefore, we propose the self-dynamic gate, which introduces the gating mechanism first to filter the feature and introduce the filtered feature to fuse them dynamically.

3. Methods

First, we introduce the preliminaries of the deep-learning-based method, which is also the basic theory of our work. The deep-learning-based method assigns the pixel to one of the surrounding nine pixels by computing the relationship information between the pixel and the surrounding nine grids. Then, we introduce the details of the model design, which is an encoder–decoder architecture.

3.1. Preliminaries

As shown in Figure 1, the image F is first divided into several 16×16 grids. For every pixel p in image F, our goal is to introduce an associate map M to assign the p to one of the surrounding nine grids S_i, just as is shown in Figure 1. Mathematically, deep-learning-based methods feed the F to the network and output the associate map $M \in R^{H \times W \times 9}$. The H and W stand for the height and width. The 9 means the nine surrounding grids S. And we see the $M_s(p)$ as the probability of the pixel p belonging to the seeds S. However, there is no label to compute the loss directly. Therefore, we serve the map M as an intermediate variable to reconstruct the pixel-wise label, i.e., the property label P_g, and the location label I_g.

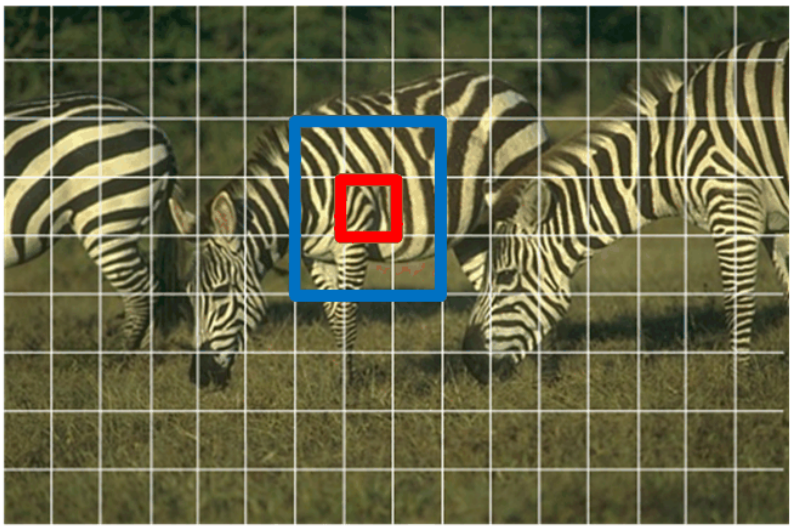

Figure 1. The image is divided into 16 × 16 grids and we compute the associate map between the pixel and surrounding nine grids.

First, we need to compute the center of the superpixel $S_c = (P_s, I_s)$, where P_s is the property vector and I_s is the location vector. The calculation can be written as:

$$P_s = \frac{\sum_{p:S\in N} \cdot P_g \cdot M_s(p)}{\sum_{p:S\in N} M_s(p)} \tag{1}$$

$$I_s = \frac{\sum_{p:S\in N} \cdot I_g \cdot M_s(p)}{\sum_{p:S\in N} M_s(p)} \tag{2}$$

where P_g and I_g are the property vector and location vector of the image F, which is also the property vector that we want to preserve. The $M_s(p)$ is the probability that the pixel p belongs to the seed S. After, we compute the property vector and location vector of the center of the superpixel S_c. We can reconstruct the property vector P_r and location vector I_r because the pixels in the superpixel have the same low- and mid-level properties. And we can compute the P_r and I_r as follows:

$$P_r = \sum_{S\in N} P_s \cdot M_s(p) \tag{3}$$

$$I_r = \sum_{S\in N} I_s \cdot M_s(p) \tag{4}$$

where N is the surrounding nine superpixels. P_r and I_r are the reconstructed property and location vectors, respectively. We can obtain the loss by computing the distance between the groundtruth property vector P_g and location vector I_g and the reconstructed property vector P_r and location vector I_r. The calculation of the loss can be written as:

$$L = dist(P_g, I_g) + \frac{m}{s} dist(I_g, I_r) \tag{5}$$

where L is the loss that we want to obtain. The $dist(.)$ is the loss function, and we introduce the CrossEntropy loss function. The m and s are the balance weight and superpixel sampling interval, respectively. The first part of Equation (5) can encourage the model to group the pixels with the same property. The second can help the model produce a more compact superpixel map.

3.2. Overall Architecture

To fix the problems of under-segmentation and low compactness, we design EAGNet, an enhanced atrous extractor and self-dynamic gate network. This is shown in Figure 1. First, the original input I is fed into several CNN blocks to extract the pixel feature. Then,

we concatenate them to obtain the multi-scale pixel feature. And we introduce the enhanced atrous extractor to extract the multi-scale superpixel feature. After that, we split the multi-scale superpixel feature in the channel dim to obtain the superpixel feature of different scales. Finally, we concatenate the pixel and superpixel feature to obtain the pixel–superpixel relationship information of different scales and concatenate them for the final prediction. The whole feedforward process of the overall architecture can be written as:

$$p_1, p_2, p_3, p_4 = Backbone(I) \tag{6}$$

$$p_m = concat(p_1, p_2, p_3, p_4) \tag{7}$$

where p_1, p_2, p_3, p_4 is the pixel feature of different scales. $Backbone(.)$ is the CNN backbone and I is the input. p_m is the multi-scale pixel feature and $concat(.)$ stands for the concatenate operation.

$$s_m = EAE(p_m) \tag{8}$$

$$s_1, s_2, s_3, s_4 = split(s_m) \tag{9}$$

where $EAE(.)$ is the enhanced atrous extractor, and p_m is the multi-scale pixel feature. The s_m is the multi-scale superpixel feature. s_1, s_2, s_3, s_4 is the superpixel feature of different scales. The $split(.)$ stands for the operation of splitting in the channel dim.

$$f_{sp}^i = G(p_i, s_i)\{i = 1, 2, 3, 4\} \tag{10}$$

$$F_m = concat(f_{sp}^1, f_{sp}^2, f_{sp}^3, f_{sp}^4) \tag{11}$$

$$Q = Predict(F_m) \tag{12}$$

where i is the number of different scales. f_{sp}^i means the fused feature. p_i and s_p stand for the pixel and superpixel features of different scales. G means our proposed self-dynamic gate. The $Concat()$ means the concatenate operation. F_m is the multi-scale pixel–superpixel relationship information. Q is the associate map that reconstructs the property vector. The $Predict()$ means our segmentation head. Then, we introduce the detail information of the different parts of EAGNet.

3.3. CNN Backbone

The CNN backbone is used to extract the pixel features of different scales. As shown in Figure 2, our CNN backbone is a four-stage pure CNN backbone due to the requirement of the low computation cost. We only introduce one CNN block as a stage. Every CNN block consists of three convolution layers. The first layer is a stride-2 3×3 convolution, which is used to downsample and expand the receptive field. The remaining two layers are normal 3×3 convolution. The feedforward process of the one stage can be written as:

$$f_p = f_p + Conv3(Conv2(Conv1_{s=2}(f_p))) \tag{13}$$

where $Conv1_{s=2}$ means the stride-2 3×3 convolution, and $Conv2$ and $Conv3$ are the normal 3×3 convolution. Note that we introduce a residual connection at every stage.

The whole process of the backbone can be written as:

$$p_i = Block(p_{i-1})\{i = 1, 2, 3, 4\} \tag{14}$$

where i is the stage number, p_i is the pixel feature of a different stage i, and $block$ is our CNN block.

After that, we need to concatenate them to obtain the multi-scale pixel feature. First, we need to introduce the global average pooling on p_i to adjust the dimension of p_i to the same dimension. And we concatenate them for the next step.

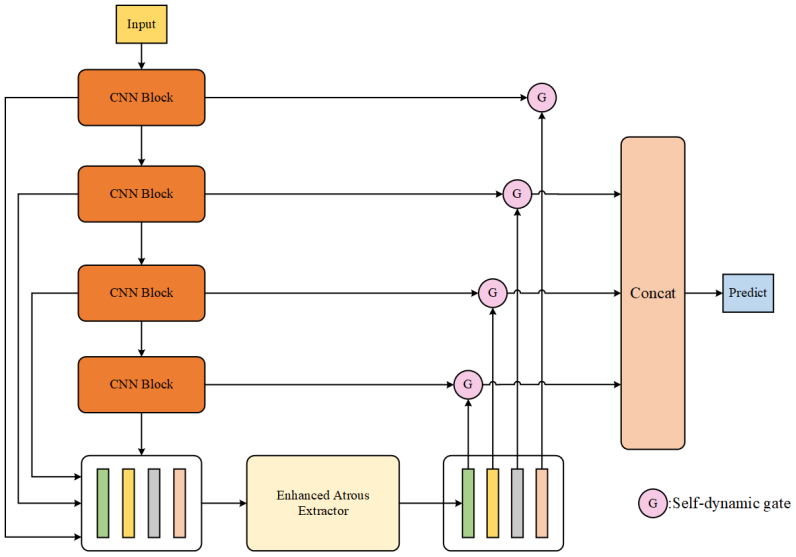

Figure 2. The overall architecture of EAGNet. The EAGNet consists of a CNN backbone, an enhanced atrous extractor, and a self-dynamic gate. The CNN backbone and the enhanced atrous extractor can extract the pixel feature and multi-scale superpixel feature. The self-dynamic gate can filter the pixel feature and superpixel feature and fuse them to obtain the pixel–superpixel relationship information for the final prediction.

3.4. Enhanced Atrous Extractor

To extract the multi-scale superpixel feature with contextual information, we design an enhanced atrous extractor (EAE). As shown in Figure 3, our proposed EAE consists of an Enhanced Atrous Module and an MLP head. The EAE Module introduces the Atrous convolution and *SiLU* function to produce the weight and sum the features to add the multi-scale superpixel feature with contextual information. And the MLP can add the non-linear complexity. The input feeds to the Enhanced Atrous Module and an MLP to extract the multi-scale superpixel feature and add the contextual information under the specific receptive field.

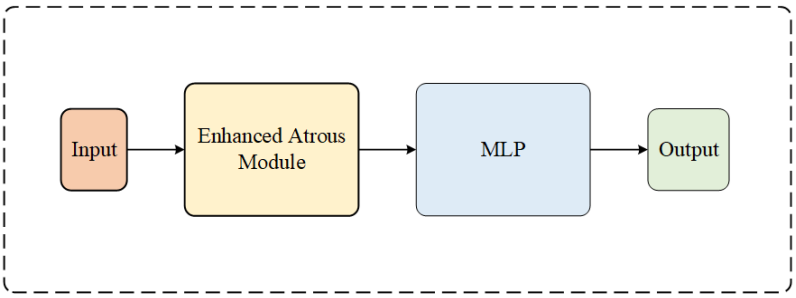

Figure 3. The overall architecture of EAE. The EAE consists of the Enhanced Atrous Module and MLP head. The Enhanced Atrous Module can extract the different superpixel features under different receptive fields.

For the Enhanced Atrous Module, as shown in Figure 4, we introduce five atrous convolutions with different dilation rates on the input to produce five superpixel features with contextual information under the specific receptive field. Then, we introduce the *SiLU* function on it to produce weights and multiply these by weighted features. Finally, we sum them up to obtain the final output. Formally, the whole process can be written as:

$$f_i = AConv_{r=i}(p_m), \{i = 1, 2, 3, 4, 5\} \quad (15)$$

where $AConv_{r=1}$ means the atrous convolution with a dilation rate equal to i, and i is the dilation rate. The f_i means the superpixel feature of dilation rate i. The P_m is the multi-scale pixel feature, which is the input of the Enhanced Atrous Module.

$$f_i = f_i \times sigmoid(f_i), \{i = 1, 2, 3, 4, 5\} \tag{16}$$

where $sigmoid(.)$ is the sigmoid non-linear activation function. And Equation (16) is also the process of the $SiLU$ activation function.

$$Output = SUM(f_i), \{i = 1, 2, 3, 4, 5\} \tag{17}$$

where $Output$ is the output of the Enhanced Atrous Module. The $SUM(.)$ is the sum operation.

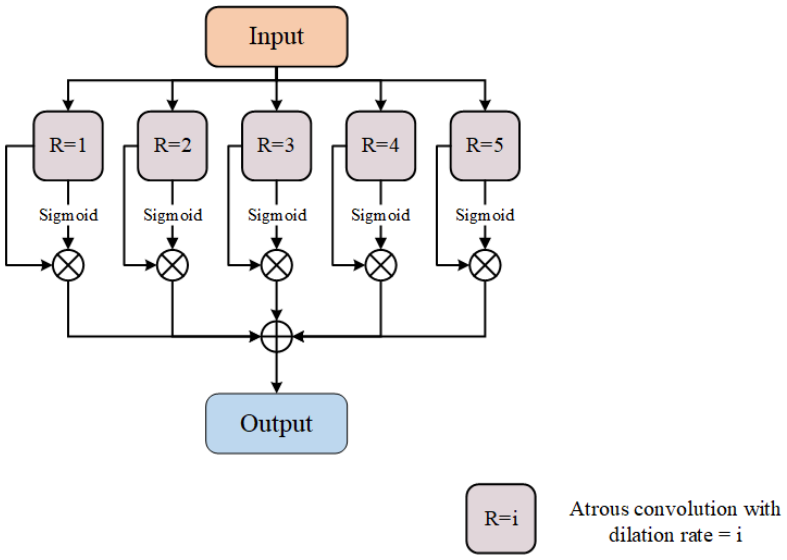

Figure 4. The overall architecture of the Enhanced Atrous Module. The Enhanced Atrous Module introduces the atrous convolution and silu function to extract the superpixel feature and strengthen them.

The Atrous convolution can extract the contextual information under the specific receptive field. The $SiLU$ function can weight the feature to strengthen the representation ability of the feature. And we can add all the features to obtain the multi-scale superpixel feature with contextual information under different receptive fields with a powerful representation ability. And we also conduct experiments to show that contextual information can improve compactness.

For the MLP part, as shown in Figure 5, it consists of two fully connected layers and a 3×3 depth-wise convolution. And every fully connected layer consists of a 1×1 convolution, a batchnorm, and a $LeakReLU$ layer. We set the expand ratio of the MLP to 2. The whole process of the MLP can be written as:

$$f_1 = fullyconnectedlayer(x) f_2 = DWConv(f_1) \tag{18}$$

$$f_2 = DWConv(f_1) \tag{19}$$

$$output = fullyconnectedlayer(f_2) \tag{20}$$

where $fullyconnectedlayer(.)$ is the fully connected layer. The DWConv is the 3×3 depth-wise convolution layer.

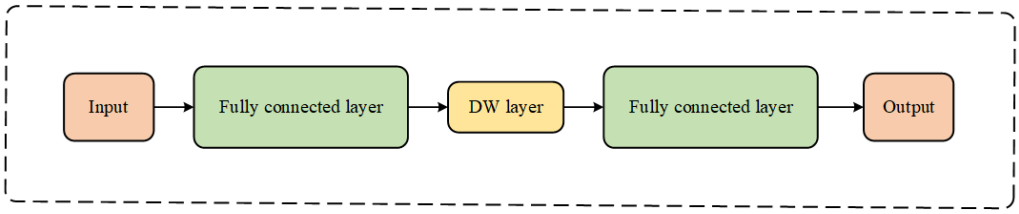

Figure 5. The overall architecture of the MLP. The MLP is the Vaillant MLP head in the transformer.

3.5. Self-Dynamic Gate

To fix the under-segmentation and inject the detail information, we design the self-dynamic gate. As shown in Figure 6, considering the requirement of the computation cost, the self-dynamic gate only has two embeddings. The embedding is a convolution layer, and we introduce the *sigmoid* on the pixel and superpixel features themselves to produce the weight, which is the gate. And we multiply the gate with the feature to filter the feature. Finally, we sum them to obtain the final output. The whole process can be written as:

$$f_p = embedding(p) \quad f_s = embedding(s) \tag{21}$$

$$f_p = f_p \times sigmoid(f_p) \quad f_s = f_s \times sigmoid(f_s) \tag{22}$$

$$Output = f_p + f_s \tag{23}$$

where *embedding* is the embedding layer. The f_p and f_s are the pixel feature and superpixel feature, respectively. The $sigmoid()$ is the *sigmoid* activation function.

The self-dynamic gate can filter and fuse the pixel and superpixel feature. And we introduce a convolution after we fuse them. And we can inject the detail information and obtain the pixel–superpixel relationship information by using the self-dynamic gate.

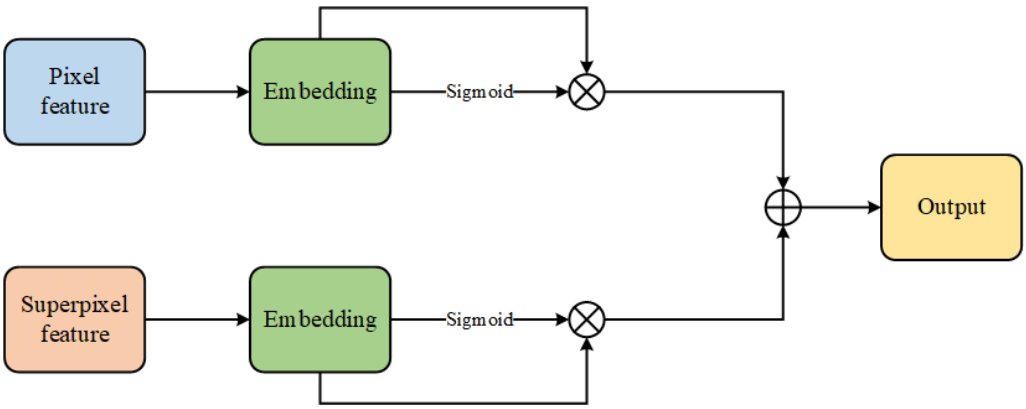

Figure 6. The overall architecture of self-dynamic gate. The embedding is the convolution layer and the sigmoid function can produce the gate by the pixel and superpixel features themselves.

4. Experiments

First, we introduce the dataset and the setting of the experiments. Then, we introduce the qualitative and quantitative results of EAGNet to prove the effectiveness and efficiency of EAGNet.

4.1. Dataset

We conducted our experiment on the Berkeley Segmentation dataset (BSDS500) [34]. BSDS500 includes 500 images, groundtruth human annotations, and benchmarking code. And we introduced 200 images for training, 100 for validation, and 200 for testing. We followed the same strategy as [19], which treats each annotation as a sample. Therefore, the training set contains 1087 images, and the test set contains 1063 images.

4.2. Implementation Detail

We implemented our method on Pytorch 1.11.0 and introduced Adam with $B_1 = 0.9$ and $B_2 = 0.999$ to train 3000 epochs. We randomly cropped the image to 208×208 as input. We set the batchsize as 16 and the learning rate as 0.00003. Moreover, the learning rate was discounted by 0.5 after 2000 epochs.

4.3. Evaluation Metrics

We chose three popular metrics to evaluate our method, which are the achievable segmentation accuracy (ASA), boundary recall (BR), and compactness (CO). ASA stands for the upper bound of the achievable segmentation accuracy. BR-BP can assess the superpixel segmentation method's ability to identify semantic boundaries. The higher scores of these metrics stand for better performance. And all the x-axes are the number of superpixels.

4.4. Comparison with State of the Arts

As shown in Figure 7, compared with the other state-of-the-art methods (e.g., ers, etps, LSC, and SLIC), the ers and etps are the methods to let the superpixel act as a differentiable, and the LSC and SLIC are the best methods of the k-means-based methods. Compared with the SLIC, SEEDs, and LSC, we can see that the ASA, BR-BP, and CO for them is marginally lower than ours. And compared with the ERS and ETPS, our method also achieves the best ASA, CO, and BR-BP. We also compare with the deep-learning-based method SCN, and we can see that our ASA and CO is higher than the SCN. The BR-BP is similar with the SCN. Therefore, our method achieves the state-of-the-art performance in the ASA, BR-BP, and CO, which means that our method can achieve the state-of-the-art performance. Our methods can segment the boundary and the part with the same color or other mid-level properties.

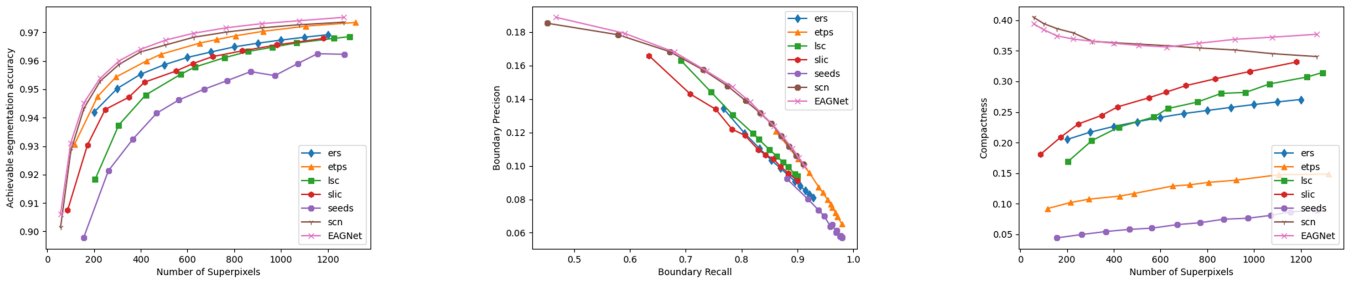

Figure 7. Comparsion with other state-of-the-art methods. The (**a**) is the ASA, (**b**) is the BR-BP, and (**c**) is the compactness.

4.5. The Visual Comparisons Results of BSDS500

As shown in Figure 8, for the first row, we can see that our methods can segment the red box of the aircraft tail accurately. The ERS and ETPS can not segment the red box of the aircraft tail accurately, which means that our method can better segment the low-level and mid-level properties. And we can see that the compactness of our method is also the best between these methods. In the second row, we can see that our method can segment the logo of the car. Other methods can not segment the logo of the car, which means that our method can segment the small object better. For the third and fourth rows, we can see that only our method can segment the hand of the child and the small window of the boat, which also proves that our method can segment the low-level and mid-level properties and small objects accurately. It proves that our method can solve the under-segmentation problem.

Figure 8. The visual results of our methods and the other state-of-the-art methods.

4.6. Ablation Study

To prove the effectiveness of every part of our proposed methods, we also conduct experiments on BSDS500. As shown in Figure 9, to prove the effectiveness of our proposed enhanced atrous extractor and self-dynamic gate, the baseline means we remove all our proposed parts. The enhanced atrous extractor and the only add gate stand for only adding the enhanced atrous extractor and the self-dynamic gate, respectively. We can see that if we add the enhanced atrous extractor, the ASA has a large increase, which proves the effectiveness of the enhanced atrous extractor. It stands for the multi-scale superpixel feature with contextual information that is necessary for the superpixel segmentation.

And we can see that if we only add the self-dynamic gate, the performance still has an increase. It means that our proposed filtered pixel and superpixel feature is good for superpixel segmentation, which proves the effectiveness of our proposed self-dynamic gate.

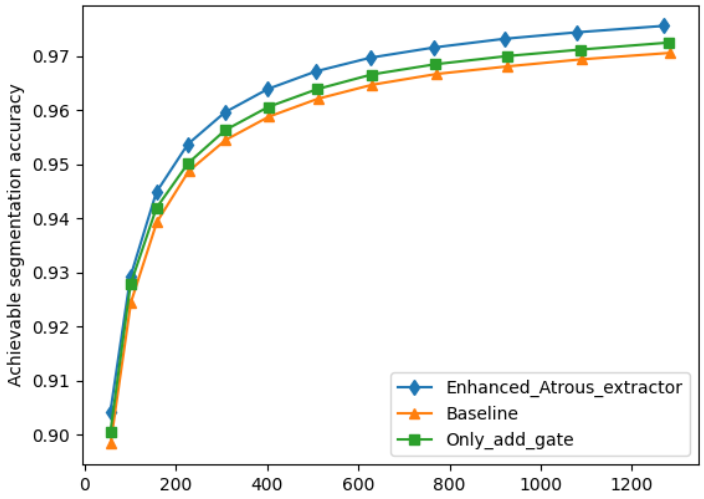

Figure 9. The ablation study of our proposed enhanced atrous extractor and self-dynamic gate.

To prove the superiority and influence of different parameters, we replace the enhanced atrous extractor with other feature extraction methods, such as the Vaillant transformer and AINet. The Vaillant transformer is the basic classic transformer without any additional modifications. The transformer of AINet can extract the superpixel feature explicitly. As shown in Figure 10a, we can see that our enhanced atrous extractor achieves the best performance. Compared with the Vaillant transformer and the transformer of AINet, we can see that our feature without contextual information results in performance degradation. And we replace the self-dynamic gate with the other feature fusion module. As shown in Figure 10c, we can see that our proposed self-dynamic gate achieves the best performance, which proves the superiority of the self-dynamic gate. And we can see if we introduce the add and multiply to fuse the feature that the performance has a large decrease, which means that our self-dynamic gate can fuse the feature better. We replace the activation function of our proposed self-dynamic gate to probe the effectiveness of different combinations of activation. As shown in Figure 10b, we can see that if we replace the sigmoid with Tanh or ReLU, the performance is similar to ours. But if we replace the sigmoid with Softmax, the performance has a large decrease, which means Softmax is harmful to the filtering features.

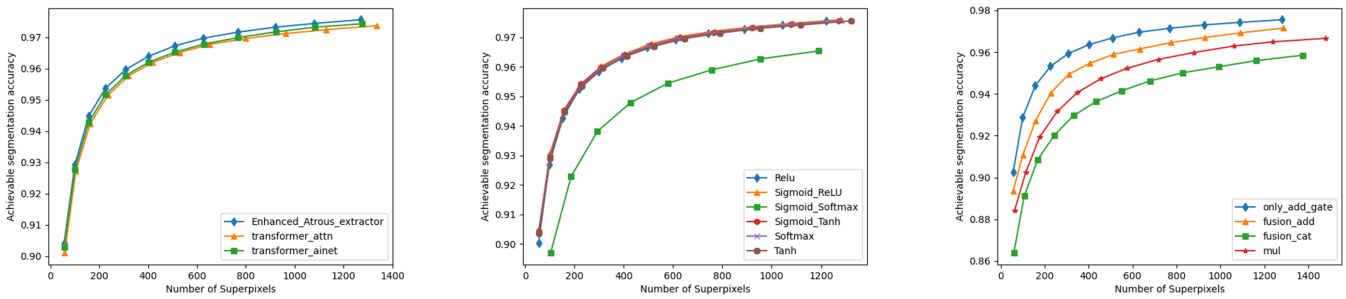

Figure 10. The ablation study of different settings.

4.7. More Discussion

With the development of remote sensing and deep-learning technology [35–39], the characteristics of remote sensing images, such as the large dimension and number

of objects, which often result in the huge computational cost, make it hard to meet the needs of real-world application. Typical feature analysis [40–42], road extraction [43], urban planning [44], and other practical applications are of great civil and military significance. The traditional segmentation algorithm can only extract low-level features, which can not meet the requirements of high-resolution remote sensing image segmentation. In order to prove that the proposed EAGNet can reduce the number of primitives, we introduce the EAGNet on remote sensing images to process them.

First, we use some examples of remote sensing images. As shown in Figure 11, we chose some images from the UCM dataset. These images have complex scenarios and different feature characters. And the UCM dataset has 22 classes with 100 images in each class. Every class stands for the most common scene in the real world. As shown in Figure 12, we introduced the EAGNet on the remote sensing images. We can see the buildings of complex scenarios are segmented accurately, which means our proposed EAGNet can reduce the primitives by seeing the superpixel as one pixel with high generalization. It stands that our EAGNet can reduce computation costs to meet the demand of the real-world remote sensing application.

Figure 11. Examples of the remote sensing dataset UCM. We can see that the UCM dataset consists of the most common scenes in the real world.

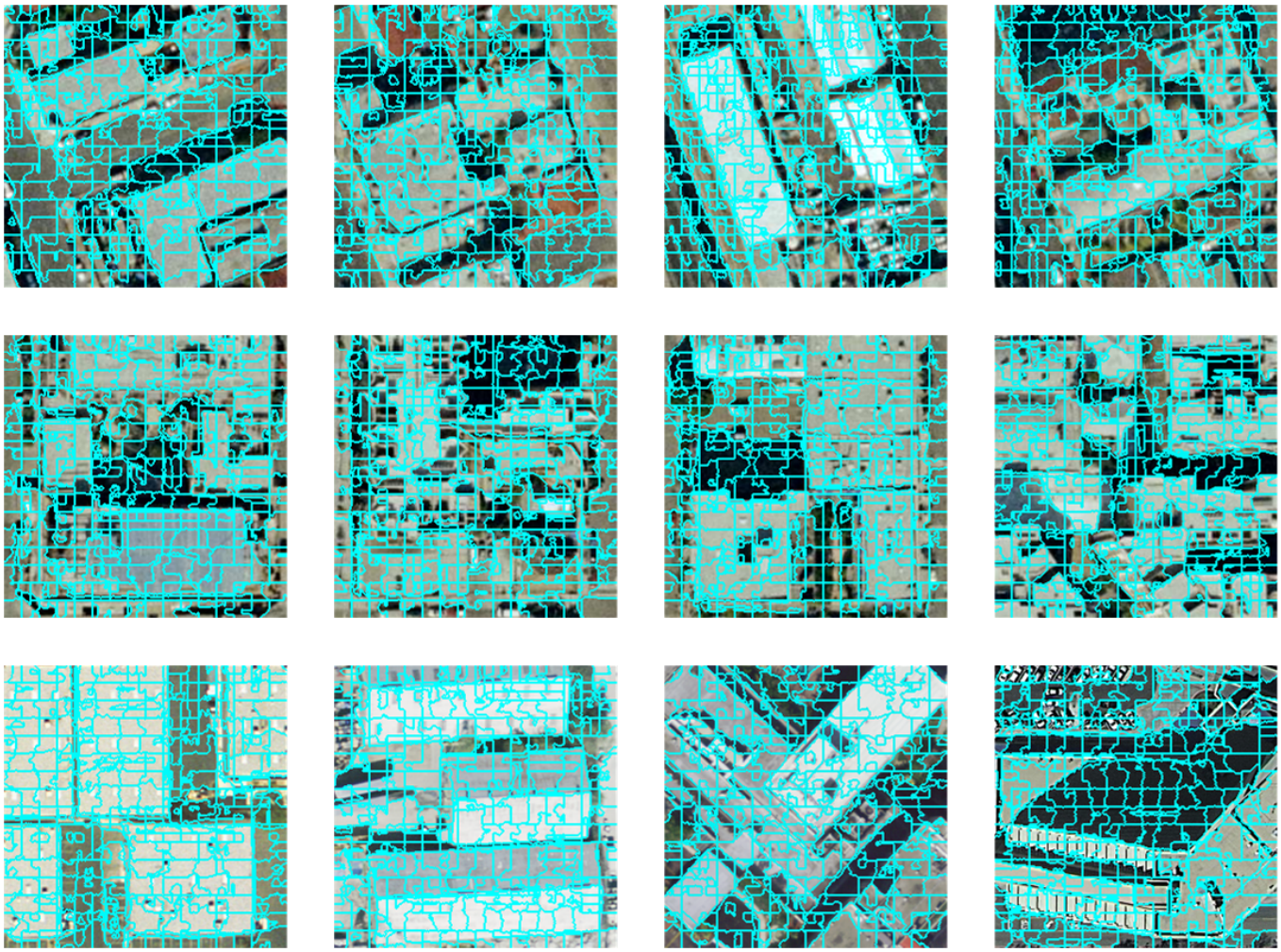

Figure 12. The visual result of EAGNet on remote sensing images.

5. Conclusions

We proposed EAGNet, which consists of an enhanced atrous extractor and self-dynamic gate. The enhanced atrous extractor can extract the multi-scale superpixel feature with contextual information and the self-dynamic gate can filter and fuse the feature effectively. EAGNet can solve under-segmentation effectively. And we conducted massive experiments to show that our methods can achieve 97.61 in ASA and 18.85 in CO of the BSDS500 and can be applied in the remote sensing fields. And we will reduce the computation complexity and explore more applications of superpixels in the remote sensing field.

Author Contributions: Methodology and conceptualization, B.L.; validation, Z.Z.; writing, T.H.; formal analysis, H.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The BSDS500 dataset and the reference codes in this work are available at https://github.com/davidstutz/superpixel-benchmark (accessed on 30 November 2023).

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

Abbreviations

The following abbreviations are used in this manuscript:

SLIC	Simple Linear Iterative Clustering
SNIC	Simple Non-Iterative Clustering
LSC	Linear Spectral Clustering
GCN	Graph Convolutional Network
CNN	Convolution Neural Network
SSN	Sampling Superpixel Network
SCN	Superpixel Fully Connected Network

References

1. Zhang, H.; Lin, M.; Yang, G.; Zhang, L. ESCNet: An end-to-end superpixel-enhanced change detection network for very-high-resolution remote sensing images. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *34*, 28–42. [CrossRef] [PubMed]
2. Shi, W.; Sui, H. An effective superpixel-based graph convolutional network for small waterbody extraction from remotely sensed imagery. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *109*, 102777. [CrossRef]
3. Gu, Y.; Liu, T.; Li, J. Superpixel tensor model for spatial–spectral classification of remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 4705–4719. [CrossRef]
4. Arisoy, S.; Kayabol, K. Mixture-based superpixel segmentation and classification of SAR images. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1721–1725. [CrossRef]
5. Zhang, W.; Xiang, D.; Su, Y. Fast multiscale superpixel segmentation for SAR imagery. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 4001805. [CrossRef]
6. Qin, F.; Guo, J.; Lang, F. Superpixel segmentation for polarimetric SAR imagery using local iterative clustering. *IEEE Geosci. Remote Sens. Lett.* **2014**, *12*, 13–17.
7. Lang, F.; Yang, J.; Yan, S.; Qin, F. Superpixel segmentation of polarimetric synthetic aperture radar (sar) images based on generalized mean shift. *Remote Sens.* **2018**, *10*, 1592. [CrossRef]
8. Yin, J.; Wang, T.; Du, Y.; Liu, X.; Zhou, L.; Yang, J. SLIC superpixel segmentation for polarimetric SAR images. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5201317. [CrossRef]
9. Wang, W.; Xiang, D.; Ban, Y.; Zhang, J.; Wan, J. Superpixel segmentation of polarimetric SAR images based on integrated distance measure and entropy rate method. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 4045–4058. [CrossRef]
10. Liu, Y.; Zhang, H.; Cui, Z.; Lei, K.; Zuo, Y.; Wang, J.; Hu, X.; Qiu, H. Very High Resolution Images and Superpixel-Enhanced Deep Neural Forest Promote Urban Tree Canopy Detection. *Remote Sens.* **2023**, *15*, 519. [CrossRef]
11. Ban, Z.; Liu, J.; Cao, L. Superpixel segmentation using Gaussian mixture model. *IEEE Trans. Image Process.* **2018**, *27*, 4105–4117. [CrossRef]
12. Shen, J.; Hao, X.; Liang, Z.; Liu, Y.; Wang, W.; Shao, L. Real-time superpixel segmentation by DBSCAN clustering algorithm. *IEEE Trans. Image Process.* **2016**, *25*, 5933–5942. [CrossRef] [PubMed]
13. Xiao, X.; Zhou, Y.; Gong, Y.J.Content-adaptive superpixel segmentation. *IEEE Trans. Image Process.* **2018**, *27*, 2883–2896. [CrossRef]
14. Ren, C.Y.; Reid, I. *gSLIC: A Real-Time Implementation of SLIC Superpixel Segmentation*; Technical Report; University of Oxford, Department of Engineering: Oxford, UK, 2011; pp. 1–6.
15. Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; Susstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 2274–2282. [CrossRef] [PubMed]
16. Achanta, R.; Susstrunk, S. Superpixels and polygons using simple non-iterative clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4651–4660.
17. Li, Z.; Chen, J. Superpixel segmentation using linear spectral clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1356–1363.
18. Jampani, V.; Sun, D.; Liu, M.Y.; Yang, M.H.; Kautz, J. Superpixel sampling networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 352–368.
19. Yang, F.; Sun, Q.; Jin, H.; Zhou, Z. Superpixel segmentation with fully convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13964–13973.
20. Dollár, P.; Zitnick, C.L. Structured forests for fast edge detection. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 1841–1848.
21. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
22. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhei, X.; Unterthinder, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.

23. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
24. Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 568–578.
25. Zhang, J.; Peng, H.; Wu, K.; Liu, M.; Xiao, B.; Fu, J.; Yuan, L. Minivit: Compressing vision transformers with weight multiplexing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12145–12154.
26. Ding, M.; Xiao, B.; Codella, N.; Luo, P.; Wang, J.; Yuan, L. Davit: Dual attention vision transformers. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer Nature: Cham, Switzerland, 2022; pp. 74–92.
27. Koonce, B. *EfficientNet: Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization*; Apress: Berkeley, CA, USA, 2021; pp. 109–123.
28. Huang, Z.; Xu, W.; Yu, K. Bidirectional LSTM-CRF models for sequence tagging. *arXiv* **2015**, arXiv:1508.01991.
29. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
30. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600.
31. Pham, H.X.; Bozcan, I.; Sarabakha, A.; Haddadin, S.; Kayacan, E. Gatenet: An efficient deep neural network architecture for gate perception using fish-eye camera in autonomous drone racing. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 4176–4183.
32. Li, X.; Zhao, H.; Han, L.; Tong, Y.; Yang, K. Gff: Gated fully fusion for semantic segmentation. *arXiv* **2019**, arXiv:1904.01803.
33. Shi, Z.; Shen, X.; Chen, H.; Lyu, Y. Global semantic consistency network for image manipulation detection. *IEEE Signal Process. Lett.* **2020**, *27*, 1755–1759. [CrossRef]
34. Arbelaez, P.; Maire, M.; Fowlkes, C.; Fowlkes, C.; Malik, J. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 898–916. [CrossRef]
35. Nogueira, K.; Penatti, O.A.B.; Dos, Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognit.* **2017**, *61*, 539–556. [CrossRef]
36. Zou, Q.; Ni, L.; Zhang, T.; Wang, Q. Deep learning based feature selection for remote sensing scene classification. *IEEE Geosci. Remote Sens. Lett.* **2015**, *12*, 2321–2325. [CrossRef]
37. Li, E.; Xia, J.; Du, P.; Lin, C.; Samat, A. Integrating multilayer features of convolutional neural networks for remote sensing scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 5653–5665. [CrossRef]
38. Ma, L.; Liu, Y.; Zhang, X.; Ye, Y.; Yin, G.; Johnson, B.A. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 166–177. [CrossRef]
39. Zavorotny, V.U.; Voronovich, A.G. Scattering of GPS signals from the ocean with wind remote sensing application. *IEEE Trans. Geosci. Remote Sens.* **2000**, *38*, 951–964. [CrossRef]
40. Benediktsson, J.A.; Pesaresi, M.; Amason, K. Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1940–1949. [CrossRef]
41. Camps-Valls, G.; Mooij, J.; Scholkopf, B. Remote sensing feature selection by kernel dependence measures. *IEEE Geosci. Remote Sens. Lett.* **2010**, *7*, 587–591. [CrossRef]
42. Ruiz, L.A.; Fdez-Sarría, A.; Recio, J.A. Texture feature extraction for classification of remote sensing data using wavelet decomposition: A comparative study. In Proceedings of the 20th ISPRS Congress, Istanbul, Turkey, 12–23 July 2004; Volume 35, pp. 1109–1114.
43. Wang, W.; Yang, N.; Zhang, Y.; Wang, F.; Cao, T.; Eklund, P. A review of road extraction from remote sensing images. *J. Traffic Transp. Eng.* **2016**, *3*, 271–282. [CrossRef]
44. Maktav, D.; Erbek, F.S.; Jürgens, C. Remote sensing of urban areas. *Int. J. Remote Sens.* **2005**, *26*, 655–659. [CrossRef]