Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6903130, 13 pages
https://doi.org/10.1155/2020/6903130

Research Article

Semantic Segmentation under a Complex Background for Machine Vision Detection Based on Modified UPerNet with Component Analysis Modules

Jian Huang, Guixiong Liu, and Bodi Wang

School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou 510640, China

Correspondence should be addressed to Guixiong Liu; megxliu@scut.edu.cn

Received 2 May 2020; Revised 31 July 2020; Accepted 17 August 2020; Published 12 September 2020

Academic Editor: Yang Li

Semantic segmentation with convolutional neural networks under a complex background using the encoder-decoder network increases the overall performance of online machine vision detection and identification. To maximize the accuracy of semantic segmentation under a complex background, it is necessary to consider the semantic response values of objects and components and their mutually exclusive relationship. In this study, we attempt to improve the low accuracy of component segmentation. The basic network of the encoder is selected for the semantic segmentation, and the UPerNet is modified based on the component analysis module. The experimental results show that the accuracy of the proposed method improves from 48.89% to 55.62% and the segmentation time decreases from 721 to 496 ms. The method also shows good performance in vision-based detection of 2019 Chinese Yuan features.

1. Introduction

As one of the primary tasks of machine vision, semantic segmentation differs from image classification and object detection. The image classification process involved the recognition of the the type of object but cannot provide position information [1], whereas object detection can be used to detect the boundary and type of the object but cannot provide the actual boundary information [2]. On the other hand, semantic segmentation can recognize the type of the object and divide the actual area at the pixel level, as well as implement certain machine vision detection functions, such as positioning and recognition [3]. As we start from image classification, move to object detection, and finally reach semantic segmentation, the accuracy of the output range and position information improves [4]. In the same manner, the recognition precision increases from the image-level to the pixel-level. Semantic segmentation achieves the best recognition accuracy; therefore, it is useful in (1) distinguishing the entity from the background, (2) obtaining the position information (centroid) clearly physically defined by indirect calculation, and (3) performing machine vision detection and identification organization, which require high spatial resolution and reliability [5, 6].

The online semantic segmentation with convolutional neural networks (CNNs) under a complex background is effective for improving the overall performance of online machine vision detection and identification [7] when maintaining the same architecture of the encoder-decoder network and convolutional and pooling layer and equivalently transforming the fully connected layer, thus yielding broad generalization. In recent years, ResNet has been used to replace the shallow CNN to optimize semantic segmentation results significantly [8]. For machine vision detection and identification under a random-texture complex background, it is necessary to eliminate the random-texture complex background to extract the object without affecting the original features of the object [9]. The difficulty lies in the randomness of the textured background, which makes it difficult to employ typical periodic texture elimination techniques, such as the frequency domain filtering and image matrix methods [10, 11]. On the

contrary, the encoder-decoder semantic segmentation network ultimately retains the classification components in the network backbone, thus exhibiting larger receptive fields and better pixel recognition ability [12, 13], as depicted in Figure 1. Unreasonably selected and consequently incorrectly used component analysis modules will lead to an excessively small foreground range, resulting in the misjudgment of component pixels. If the component analysis module is too sensitive, the foreground range will be too broad; thus, it would be difficult to remove misjudged pixels [14]. Therefore, in the process of semantic segmentation under the complex background, it is necessary to consider objects, the contradiction between the component semantic response values, and their mutual exclusion relationship, while maximizing the accuracy of the semantic segmentation under the complex background using the encoder-decoder network.

Figure 2 shows a flowchart of the semantic segmentation under the complex background using the encoder-decoder network. The process can be described as follows: the component classifier of the encoder-decoder network recognizes the pixel-level semantics and response of the pixels in the image; the object classifier recognizes the pixel-level object semantics and the response and extracts misjudged pixels of the foreground object in semantic segmentation; finally, the mutually exclusive relationship between component semantics and object semantics is considered, and non-background-independent semantics are determined to achieve effective semantic segmentation under a complex background to improve the model accuracy [15].

In this study, we focus on online semantic segmentation under a complex background using the encoder-decoder network to solve the above described mutual exclusion relationship problem between component semantics and object semantics. The main contributions of this study are threefold:

(i) We attempted to improve the low accuracy of component segmentation and selected the superior basic encoder-decoder network according to the performance.

(ii) We modified the UPerNet based on the component analysis module to maximize the accuracy of the semantic segmentation under a complex background using the encoder-decoder network while maintaining an appropriate segmentation time.

(iii) We show that the proposed method is superior to previous encoder-decoder network and has satisfactory accuracy and segmentation time. We also show the application of the proposed method in bill-note anticounterfeiting identification.

The rest of this paper is organized as follows. In Section 1, we outline related works. In Section 2, we introduce a method for semantic segmentation under a complex background using the encoder-decoder network. In Section 3, we verify the proposed method. In Section 4, we present the conclusions.

2. Related Work

2.1. Evaluation of the Semantic Segmentation Performance. We can generally evaluate the CNN semantic segmentation performance from the accuracy and running speed. The accuracy indicators usually include the pixel accuracy [16], mean intersection over union [16], and mean average precision [17]. The pixel accuracy PA is defined as the number of pixels segmented correctly accounting for the total number of image pixels; the mean intersection over union $\overline{IoU}$ is defined as the degree of coincidence between the segmentation results and their ground-truth; the mean average precision AP^{IoU_T} is the mean of average precision scores for segmentation results, whose intersection over union no less than IoU_T, for each classes.

If the object detected by machine vision has k categories, the semantic segmentation model requires the label of the $k+1$ categories denoted as $L = \{l_0, l_1, \ldots, l_k\}$, including the background. Denoting the number of pixels of l_i mis-recognized as the pixel of l_j and $l_i (i \neq j)$ as p_{ij} and p_{ii} respectively, the numbers of detected objects of l_i mis-recognized as l_j and $l_i (i \neq j)$ as N_{ij} and N_{ii}, respectively, the pixel accuracy can be calculated as follows:

$$\begin{aligned} \mathrm{PA} &= \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}}; \\ \overline{IoU} &= \sum_{i=0}^{K} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}}, \end{aligned} \tag{1}$$

$$\mathrm{AP}^{IoU_T} = \frac{\sum_{i=0}^{k} N_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} N_{ij}}. \tag{2}$$

The running speed of CNN semantic segmentation can be measured by indicators including the segmentation time T_{seg} [18], which is defined as the time needed to segment the image by running the algorithm. The theoretically shortest possible time required to segment the image is also labeled as the theoretical segmentation time $T_{\mathrm{seg-t}}$, and the time required for the algorithm to actually segment the image is known as the actual segmentation time $T_{\mathrm{seg-a}}$. If not otherwise specified, $T_{\mathrm{seg-a}}$ is denoted as T_{seg}.

2.2. End-To-End Encoder-Decoder Semantic Segmentation Framework. Although CNN semantic segmentation performs as a single-step end-to-end process, which is not further divided into multiple modules to deal with, the connection of numerous modules directly affects the CNN. The end-to-end semantic segmentation framework using the encoder-decoder enables the CNN to detect images with any resolution and output prediction map results with constant resolution. Typical networks include fully convolutional networks (FCN) [19], SegNet [20], and U-Net [21].

Figure 3 shows a schematic of the FCN model. The FCN is an end-to-end semantic segmentation framework

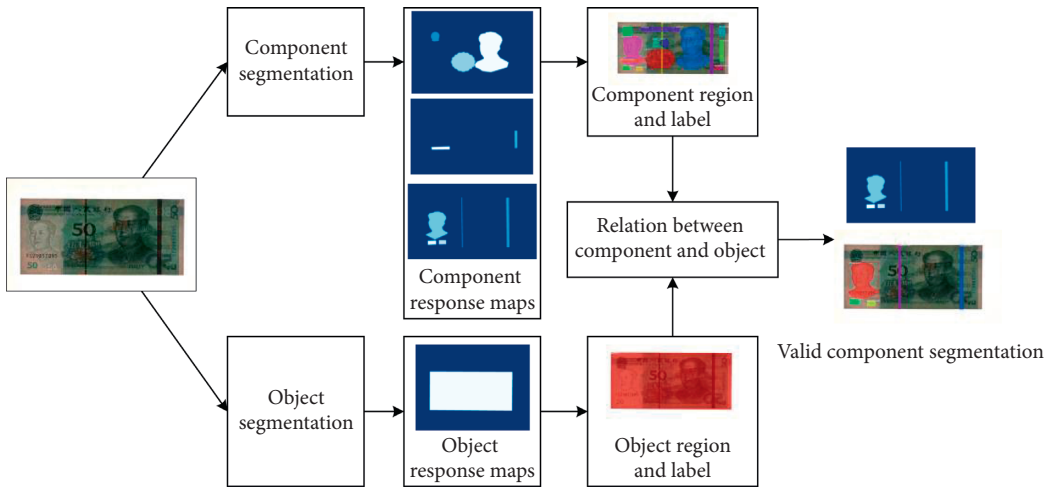


Figure 1: Flowchart of semantic segmentation under the complex background using encoder-decoder network.

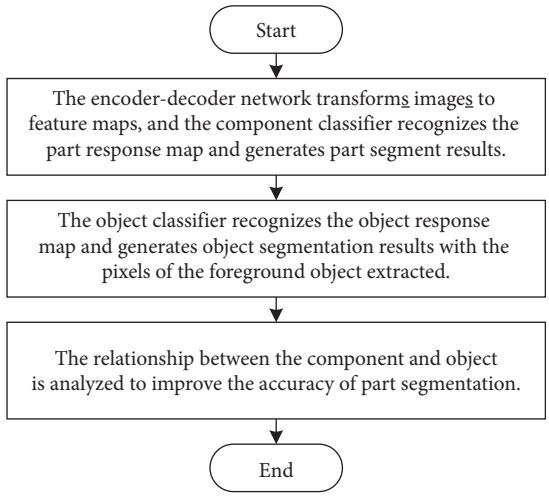


Figure 2: Flowchart of the semantic segmentation under a complex background using the encoder-decoder network.

proposed by Jonathan Long et al. (University of California, Berkeley) in 2014. The main idea is as follows: the operation of a fully connected layer is equivalent to the convolution of a feature map and a kernel function of identical size. The fully connected layer is converted into a convolution layer, which converts the CNN into a full convolution operation network consisting of a complete convolution layer (convolution operation) and pooling layer (convolution operation) to process images of any resolution. In this manner, the limitation of the fully connected layer is overcome, i.e., images with different resolutions can be processed. The original resolution is restored after eight times bilinear upsampling by taking the pooling layer as an encoder, designing a cross-layer superimposed architecture as a decoder, yielding the final output feature map of the network by upsampling, and adding to the output feature map of each pooling layer (namely, the encoder) to obtain a feature map with higher resolution. The CNN can perform end-to-end semantic segmentation through a fully convolutional and cross-layer superimposed architecture; therefore, various CNNs are capable of achieving end-to-end semantic segmentation. Using the framework described, the $\overline{IoU}$ reached 62.2% in the VOC2012 semantic segmentation testing set, which is 10.6 % higher than the classic methods and 12.2% (its $\overline{IoU}$ is 50.0%) higher than the SDS [22] further segmented by CNN object detection and classical method.

The ResNet proposed by the Amazon Artificial Intelligence Laboratory serves as a basic network for constructing FCNs for semantic segmentation; the $\overline{IoU}$ in VOC2012 reaches 8.6% [23]. The prediction results of the FCN application are obtained by eight-fold bilinear interpolation of the feature map, including the problems of detail loss, smoothing of complex boundaries, and poor detection sensitivity of small objects. The results ignore the global scale of the image, possibly exhibiting regional discontinuity for large objects that exceed the receptive field. Incorporating full connection and upsampling increase the size of the network and introduces a large number of parameters to be learned.

Figure 4 shows a schematic of the SegNet model, which is an efficient, real-time end-to-end semantic segmentation network proposed by Alex Kendall et al. (Cambridge University) in 2015. The idea is that the encoder and the decoder have a one-to-one correspondence, and the network applies the pooled index in the encoder's maximum pooling to perform nonlinear upsampling, thus forming a sparse feature map; then, it performs convolution to generate a dense feature map. SegNet defines the basic network of the encoder-decoder and deletes the fully connected layer to generate global semantic information. The decoder utilizes the encoder information without training, while the required amount of training parameters is 21.7% of that of the FCN. For the prediction of the results, SegNet and FCN occupy a GPU memory of 1052 and 1806 MB, respectively, and the GPU memory occupancy on GPU GTX 980 (video memory 4096 MB) is 25.68% and 44.09%, respectively. Therefore, the occupancy of SegNet is 18.41% lower than that of FCN. In [20], the design of SegNet on ResNet was described, and the $\overline{IoU}$ in VOC2012 reached 80.4% [24]. The $\overline{IoU}$ of SegNet tested in VOC2012 was reported to be 59.9%, and the

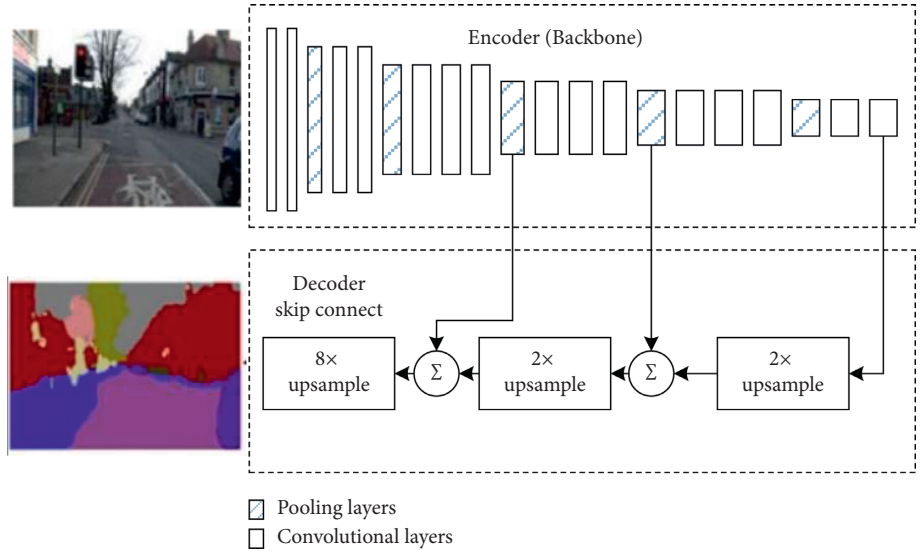


Figure 3: Schematic of FCN model.

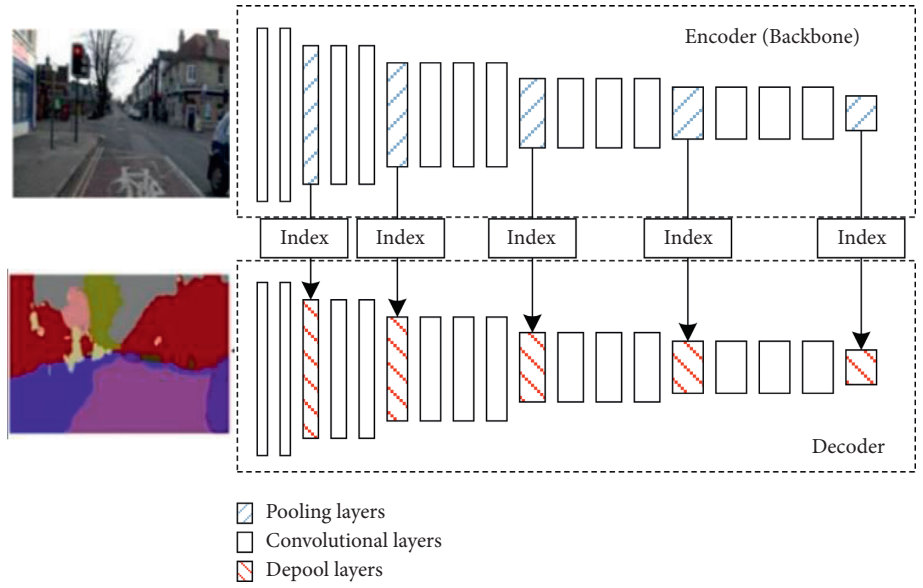


Figure 4: Schematic of SegNet model.

efficiency was found to be 2.3% lower than that of FCN; furthermore, there was the problem of false detection at the boundary.

Figure 5 shows a schematic of the U-Net model, which was proposed by Olaf Ronneberger (University of Freiburg, Germany) in 2015. The idea was to design a basic network that can be trained by semantic segmentation images and modify the FCN cross-layer overlay architecture with the high-resolution feature map channels retained in the upsampling section and then connect it to the decoder output feature map in the third dimension. Furthermore, a tiling strategy without limited by GPU memory was proposed; with this strategy, a seamless semantic segmentation of arbitrary high-resolution images was achieved. With U-Net, a $\overline{IoU}$ of 92.0% and 77.6% was achieved in the grayscale image semantic segmentation datasets PhC-U373 and DIC-HeLa, respectively. The skip connection was used in the ResNet framework to improve U-Net, and a $\overline{IoU}$ of 82.7% was achieved in the VOC2012 [25]. There are two key problems with the application of U-Net: the basic network needs to be trained, and it can only be applied to specific task, i.e., it has poor universality.

Figure 6 shows a schematic of the UPerNet model, which was proposed by Tete Xiao (Peking University, China) in 2018. In the UPerNet framework, a feature pyramid network (FPN) with a pyramid pooling module (PPM) is appended on the last layer of the backbone network before feeding it into the top-down branch of the FPN. Object and part heads are attached on the feature map and are fused by all the layers put out by the FPN.

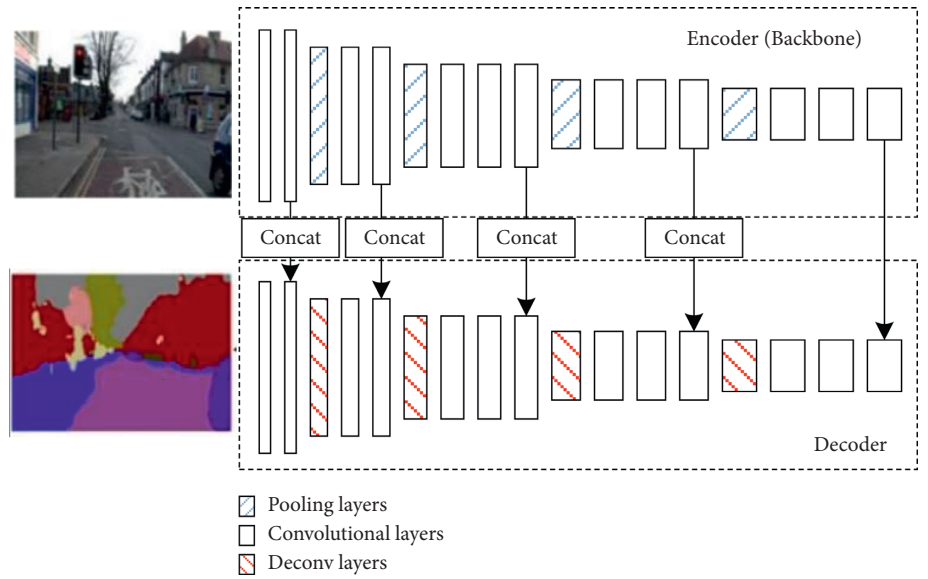


Figure 5: Schematic of U-Net model.

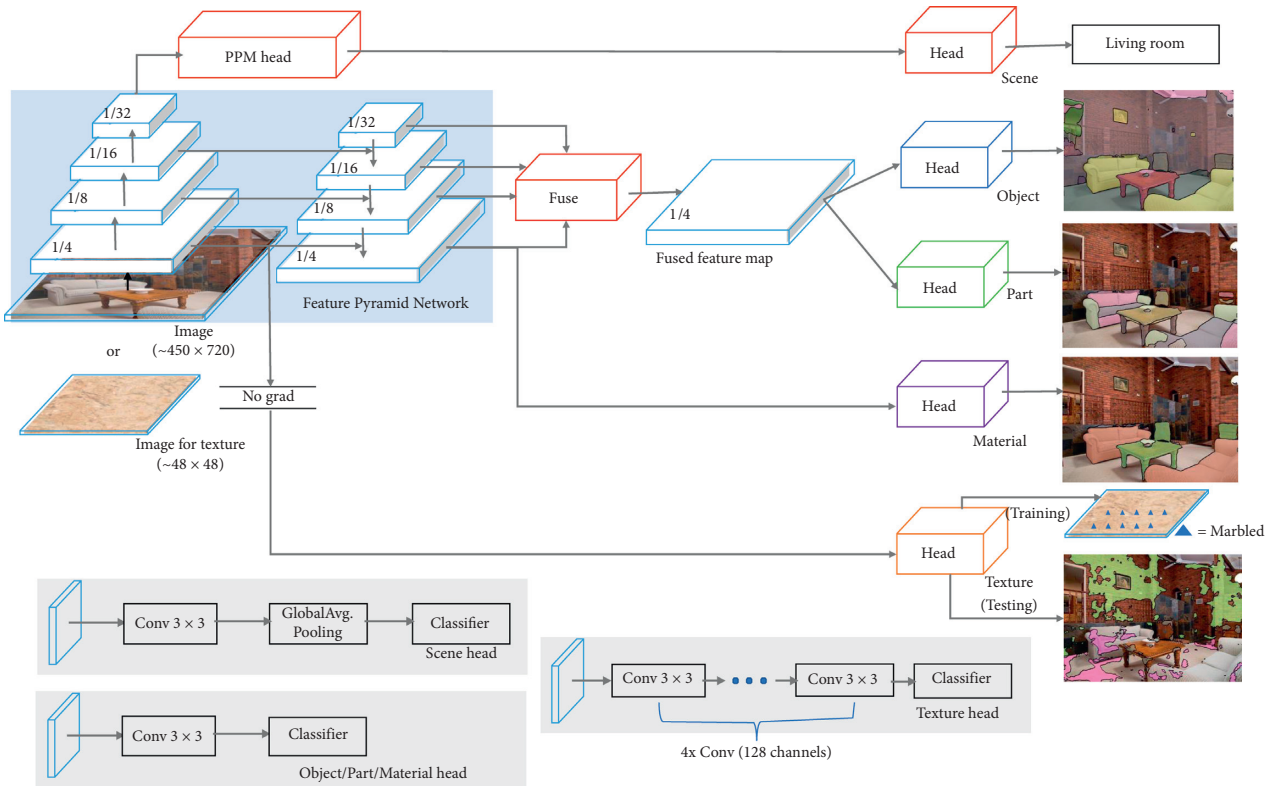


Figure 6: Flowchart of UPerNet model.

3. Material and Methods

The semantic segmentation under a complex background based on the encoder-decoder network will establish an optimized mathematical model with minimal segmentation time $T_{seg-min}$, segmentation time T_{seg}, and accuracy PA. Under the encoder-decoder network, the backbone network η_{main}, the depth d_{main}, and the decoder $\eta_{decoder}$ are obtained to form an encoder. By selecting the relatively better η_{main} and $\eta_{decoder}$ of the basic network, the component analysis module to improve the optimized architecture is proposed, and the encoder-decoder network with optimized PA for semantic segmentation under a complex background is obtained. In the encoder-decoder network, the encoder

transforms color images (three 2D arrays) to 2048 2D arrays. The encoder is composed of convolutional layers and pooling layers, and it could be trained on large-scale classification datasets, such as ImageNet, to gain greater feature extraction capability.

Modeling of semantic segmentation under a complex background using the encoder-decoder network and selection of η_{main} and η_{decoder}.

The encoder network is determined by the backbone network η_{main}, depth d_{main}, and decoder η_{decoder}. Segmentation time T_{seg} and accuracy PA depend on η_{decoder}, η_{main}, and d_{main}, which can be expressed as $\text{PA}(\eta_{\text{decoder}}, \eta_{\text{main}}, d_{\text{main}})$ and $T_{\text{seg}}(\eta_{\text{decoder}}, \eta_{\text{main}}, d_{\text{main}})$. Denoting the minimal segmentation time as $T_{\text{seg-min}}$ (the recommended value is 600 ms), the mathematical model of the optimization for semantic segmentation under a complex background based on the encoder-decoder network is as follows:

$$\begin{cases} \max & \text{PA}(\eta_{\text{decoder}}, \eta_{\text{main}}, d_{\text{main}}), \\ \text{s.t.} & T_{\text{seg}}(\eta_{\text{decoder}}, \eta_{\text{main}}, d_{\text{main}}) \le T_{\text{seg-min}}. \end{cases} \tag{3}$$

The parameters of the model to be optimized are d_{main}, η_{main}, and η_{decoder}.

First, d_{main}, η_{main}, and η_{decoder} are combined. Then, the object segmentation accuracy PA_{obj}, component segmentation accuracy PA_{comp}, and T_{seg} are compared to select the relatively better η_{main} and η_{decoder} for the basic network.

The ADE20K dataset, which has diverse annotations of scenes, objects, parts of objects, and parts of parts [26], is selected. In this paper, we denote parts of objects as component. Using a GeForce GTX 1080Ti GPU and the training method described in [27], we obtained PA_{obj} and PA_{comp} for improved FCN [19], PSPNet [28], UPerNet [29], and other major encoder-decoder networks for semantic segmentation used in the ADE20K [26] object/component segmentation dataset. We evaluated T_{seg} of different network on the ADE20K test set, which consist of 3000 different resolution images with average image size of 1.3 million pixels. Table 1 displays the pixel accuracy and segmentation time of the main network architectures on ADE20K object/component segmentation tasks, where the relatively better indices are indicated by a rectangular contour.

From Table 1, the following observation can be made. ① In all networks, PA_{comp} is less than PA_{obj} by about 30%; ② η_{main} and d_{main} are equal in networks 1, 2, and 3; PA_{comp} and PA_{obj} are better in η_{decoder} = FPN + PPM compared to η_{decoder} = FCN or η_{decoder} = PPM; ③η_{main} and η_{decoder} are equal in networks 3 and 4. When d_{main} is doubled, PA_{comp} improves slightly and T_{seg} improves significantly. After a comprehensive consideration, we selected the UPerNet [23] encoder-decoder network, where η_{main} = ResNet, d_{main} = 50, and η_{decoder} = PPM + FPN.

Figure 7 shows the architecture of semantic segmentation under a complex background implemented by UPerNet [29]. The encoder ResNet reduces the feature map resolution by 1/2 at each stage. The resolution of the output feature maps within five stages is respectively reduced to 1/2, 1/4, 1/8, 1/16, and 1/32. The decoder is PPM + FPN. Through pooling layers with different strides, the feature maps are analyzed in a multiscale manner within PPM. Through three transposed convolution layers, the resolution of the feature maps is increased two times to 1/16, 1/8, and 1/4. The upsampling restores the feature map resolution to 1/1. The component analysis module recognizes the feature map and outputs both the object/component segmentation results.

Figure 8 shows the component analysis module of UPerNet. The module is composed of the object classifier, component classifier, and component analysis module. The input of each classifier is a 1 : 1 feature map. The object classifier implements the semantic recognition of N_{Obj} kinds of objects and outputs the object probability vector $\mathbf{p}_{\text{Obj}}^{u,v}$ and the object label $C_{\text{Obj}}^{u,v}$. The component classifier implements the semantic recognition of N_{Comp} kinds of components and outputs the component probability vector $\mathbf{p}_{\text{Comp}}^{u,v}$ and the component label $C_{\text{Comp}}^{u,v}$. According to $C_{\text{Obj}}^{u,v}$ and the component object set $\mathbb{C}_{\text{Obj-Things}}$, the component analysis module only segments the $C_{\text{Comp}}^{u,v}$ that satisfies $C_{\text{Obj}}^{u,v} \in \mathbb{C}_{\text{Obj-Things}}$ and outputs the valid component label $\widehat{C}_{\text{Comp}}^{u,v}$. UPerNet outputs the object segmentation result (the object label $C_{\text{Obj}}^{u,v}$) and the component segmentation result (the valid component label $\widehat{C}_{\text{Comp}}^{u,v}$).

The component analysis module of UPerNet can be expressed as follows:

$$\begin{aligned} \widehat{C}_{\text{Comp}}^{u,v} &= f_{\text{Op}}\left(C_{\text{Obj}}^{u,v}, C_{\text{Comp}}^{u,v}, \mathbb{C}_{\text{Obj-Things}}\right) \\ &= \begin{cases} 1 \times C_{\text{Comp}}^{u,v}, & C_{\text{Obj}}^{u,v} \in \mathbb{C}_{\text{Obj-Things}}, \\ 0 \times C_{\text{Comp}}^{u,v}, & C_{\text{Obj}}^{u,v} \notin \mathbb{C}_{\text{Obj-Things}}. \end{cases} \end{aligned} \tag{4}$$

A greater PA_{comp} of $\widehat{C}_{\text{Comp}}^{u,v}$ leads to a higher component segmentation efficiency.

Equation (4) outputs $\widehat{C}_{\text{Comp}}^{u,v}$ that satisfies $C_{\text{Obj}}^{u,v} \in \mathbb{C}_{\text{Obj-Things}}$. By identifying deviations of $C_{\text{Obj}}^{u,v}$ due to the relationship between $C_{\text{Comp}}^{u,v}$ and $C_{\text{Obj}}^{u,v}$, the optimized component analysis module can improve the efficiency of component segmentation; it both meets the requirement of $T_{\text{seg-min}}$ and improves PA_{comp}.

Improvements of UPerNet for semantic segmentation under a complex background based on the component analysis module.

In this subsection, we describe the derivation of the component analysis module, the optimization of the function expression of the module, and the construction of the architecture of the component analysis module.

As shown in Figure 8, the component classifier recognizes N_{comp} component semantics and outputs the component labels $C_{\text{Comp}}^{u,v}$ of the pixel with image position (u, v) and the probability vector $\mathbf{p}_{\text{Comp}}^{u,v}$ corresponding to the various component labels. The relationship between $C_{\text{Comp}}^{u,v}$ and $\mathbf{p}_{\text{Comp}}^{u,v}$ [31] is as follows:

$$C_{\text{Comp}}^{u,v} = \text{argmax}_k p_{\text{Comp}-k}, \quad k = 1, 2, \ldots, N_{\text{Comp}}. \tag{5}$$

From equation (4) and (5), we obtain

Table 1: Pixel accuracy and segmentation time of the main network architectures on ADE20K object/component segmentation task. The rectangular contour indicates the best indices.

	Network	Backbone η_{main}	Backbone depth d_{main}	Decoder η_{decoder}	Object segmentation accuracy (%) PA_{obj}	Component segmentation accuracy (%) PA_{comp}	Segmentation time (ms) T_{seg}
1	FCN [30]	ResNet	50	FCN	71.32	40.81	333
2	PSPNet [28]	ResNet	50	PPM	80.04	47.23	483
3	UPerNet [29]	ResNet	50	PPM + FPN	80.23	48.30	496
4	UPerNet [29]	ResNet	101	PPM + FPN	81.01	48.71	604

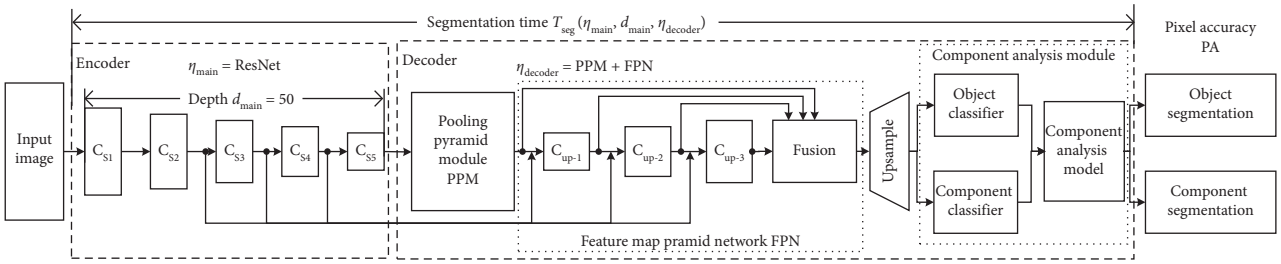


Figure 7: Flowchart of semantic segmentation under a complex background implemented by UPerNet.

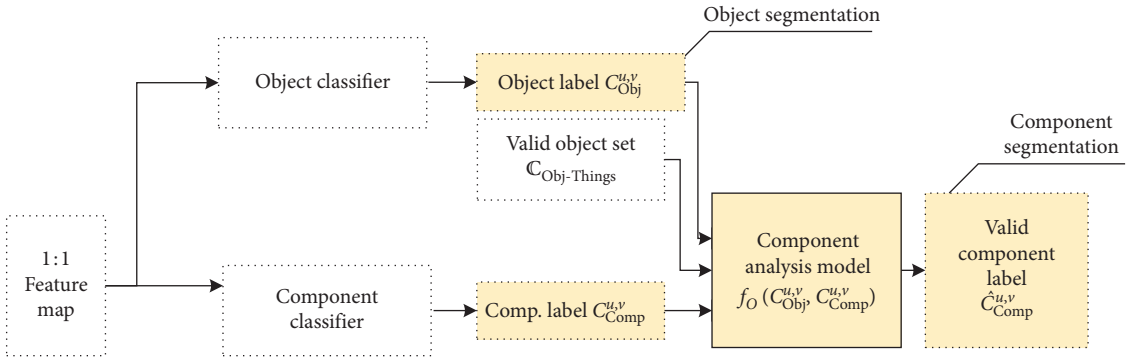


Figure 8: Flowchart of component analysis module of UPerNet.

$$\begin{aligned}\widehat{C}^{u,v}_{\text{Comp}} &= f_{\text{Op}}\left(C^{u,v}_{\text{Obj}}, C^{u,v}_{\text{Comp}}, \mathbb{C}_{\text{Obj-Things}}\right)\\ &= \begin{cases} 1 \times \text{argmax}_k p_{\text{Comp}-k}, & C^{u,v}_{\text{Obj}} \in \mathbb{C}_{\text{Obj-Things}},\\ 0 \times \text{argmax}_k p_{\text{Comp}-k}, & C^{u,v}_{\text{Obj}} \notin \mathbb{C}_{\text{Obj-Things}}, \end{cases} \quad k = 1, 2, \ldots, N_{\text{Comp}},\end{aligned} \tag{6}$$

where $p_{\text{Obj}-j}$ is the probability of $C^{u,v}_{\text{Comp}}$. Weighting $p_{\text{Obj}-j}$ over $p_{\text{Comp}-k}$ to get $\widehat{p}_{\text{Comp}-k}$ instead of $1 \times \text{argmax}_k p_{\text{Comp}-k}$, reducing the weight of low-probability object labels, and increasing PA_{comp}. With $C^{u,v}_{\text{Obj}} \in \mathbb{C}_{\text{Obj-Things}}$, if $C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{\text{Obj}}}_{\text{Comp-Obj}}$, letting $\widehat{p}_{\text{Comp}-k} = 0$ can increase the detection rate of background pixels. Therefore, the module can be expressed as follows:

$$\begin{cases} \widehat{C}^{u,v}_{\text{Comp}} = f_{\text{Op}}\left(C^{u,v}_{\text{Obj}}, C^{u,v}_{\text{Comp}}, \mathbb{C}_{\text{Obj-Things}}, \mathbb{C}^{C_{Obj}}_{\text{Comp-Obj}}\right) = \text{argmax}_k \widehat{p}_{\text{Comp}-k}, \quad k = 1, 2, \ldots, N_{\text{Comp}}, \\ \widehat{p}_{\text{Comp}-k} = \begin{cases} \left(\sum_{j=1}^{j\in\mathbb{C}_{\text{Obj-Things}} \wedge k\in\mathbb{C}^{j}_{\text{Comp-Obj}}} p_{\text{Obj}-j} \right), & k < N_{Comp} \\ \left(1 - \sum_{j=1}^{j\notin\mathbb{C}_{\text{Obj-Things}} \vee k\notin\mathbb{C}^{j}_{\text{Comp-Obj}}} p_{\text{Obj}-j} \right) p_{\text{Comp}-k}, & k = N_{\text{Comp}}, \end{cases} \quad j = 1, 2, \ldots, N_{\text{obj}}, \end{cases} \tag{7}$$

which is the component analysis module yielded by replacing $1 \times \text{argmax}_k p_{\text{Comp}-k}$ with $\text{argmax}_k \widehat{p}_{\text{Comp}-k}$ and considering $C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{\text{Obj}}}_{\text{Comp-Obj}}$.

The optimized architecture of the UPerNet component analysis module is proposed based on equation (7). Figures 9(a)–9(c) show the optimized architecture obtained by replacing $1 \times \text{argmax}_k p_{\text{Comp}-k}$ with $\arg\max_k \widehat{p}_{\text{Comp}-k}$ by considering $C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{\text{Obj}}}_{\text{Comp-Obj}}$ and by both replacing $1 \times \text{argmax}_k p_{\text{Comp}-k}$ with $\text{argmax}_k \widehat{p}_{\text{Comp}-k}$ and considering $C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{\text{Obj}}}_{\text{Comp-Obj}}$ in the component analysis module, respectively.

3.1. Experimental Results

3.1.1. ADE20K Component Segmentation Task. For the UPerNet model, the backbone network of the encoder was ResNet, $d_{\text{main}} = 50$, and the decoders are PPM + FPN + component analysis modules (before/after modification). We trained each network on the object/component segmentation task dataset ADE20K [26] to demonstrate the pixel accuracy $\text{PA}_{\widehat{\text{Part}}}$ and segmentation time T_{seg}. The experiments were run on a GeForce GTX 1080Ti GPU.

Table 2 reports $PA_{\widehat{\text{Part}}}$ and T_{seg} of the UPerNet obtained with different component analysis modules in ADE20K component segmentation task. From the results, the following observations can be made:

(i) The pixel accuracy of ResNet ($d_{\text{main}} = 50$) + PPM + FPN + the proposed modified component analysis modules with different settings increased from 48.30% (without component analysis modules) to 54.03%, 55.13%, and 55.62% while the segmentation time lengthened marginally from 483 to 492, 486, and 496 ms, respectively.

The UPerNet with modified component analysis modules showed significantly high segmentation performance. Both $\text{PA}_{\widehat{\text{Part}}}$ and T_{seg} outperformed the UPerNet with a deeper d_{main}; $\text{PA}_{\widehat{\text{Part}}}$ and T_{seg} of the architecture ($d_{\text{main}} = 50$) are 55.62% and 496 ms, while those of the architectures with no modification with $d_{\text{main}} = 101$ and 152 were 48.71% and 598 ms and 48.89% and 721 ms, respectively, as shown in Figure 9(c).

3.1.2. CITYSCAPES Instance-Level Semantic Labeling Task. We trained each UPerNet (with/without component analysis module) on the instance-level semantic labeling task of the CITYSCAPES dataset [32]. To assess the instance-level performance, CITYSCAPES uses the mean average precision AP and average precision $\text{AP}_{0.5}$ [32]. We also report the segmentation time of each network run on a GeForce GTX 1080Ti GPU and an Intel i7-5960X CPU. Table 3 presents the performances of different methods on a CITYSCAPES instance-level semantic labeling task. Table 4 presents the mean average precision AP on class-level of the UPerNet with/without the component analysis module in the CITYSCAPES instance-level semantic labeling task. From the table, it can be seen that the modified component analysis modules effectively improved the performance of the UPerNet. With the component analysis module, both AP and $\text{AP}_{0.5}$ are improved, and the segmentation time T_{seg} increased slightly from 447 to 451 ms. Most of the UPerNet AP on class-level are improved. Figure 10 shows some CITYSCAPES instance-level semantic labeling results obtained with the UPerNet with/without component analysis module.

Taking banknote detection as an example, we set up the semantic segmentation model by the component analysis modules (before/after modification) to vision-based detection of 2019 Chinese Yuan (CNY) feature in the backlight to demonstrate the segmentation performance of the proposed method.

The vision-based detection system consisted of an MV-CA013-10 GC industrial camera, an MVL-HF2528M-6MP lens, and a LED strip light. The field of view was 18.33°, and the resolution was 1280×1024. Under the backlight, we collected 25 CNY images of various denomination fronts and backs at random angles. Then, we marked four types of light-transmitting anticounterfeiting features, namely, security lines, pattern watermarks, denomination watermarks, and Yin-Yang denominations. All four features were detected in the CNY images to generate our dataset (200 images). We trained the model with different component analysis modules from our dataset to demonstrate $\text{PA}_{\widehat{\text{Part}}}$ and T_{seg}. Table 3 presents the pixel accuracy and segmentation time of UPerNet with different component analysis modules for CNY anticounterfeit features via vision-based detection, and Figure 11 shows the segmentation results of the anticounterfeiting features detected by UPerNet with/without the component analysis module.

From Table 5, it can be seen that the proposed method improved $\text{PA}_{\widehat{\text{Part}}}$ from 90.38% to 95.29% T_{seg} from 490 to 496 ms. Moreover, $\text{AP}^{IoU_T=0.5}$ increased from 96.1% to 100%, detecting all the light transmission anti-counterfeiting features without false detection, missing detection, or repeated detection.

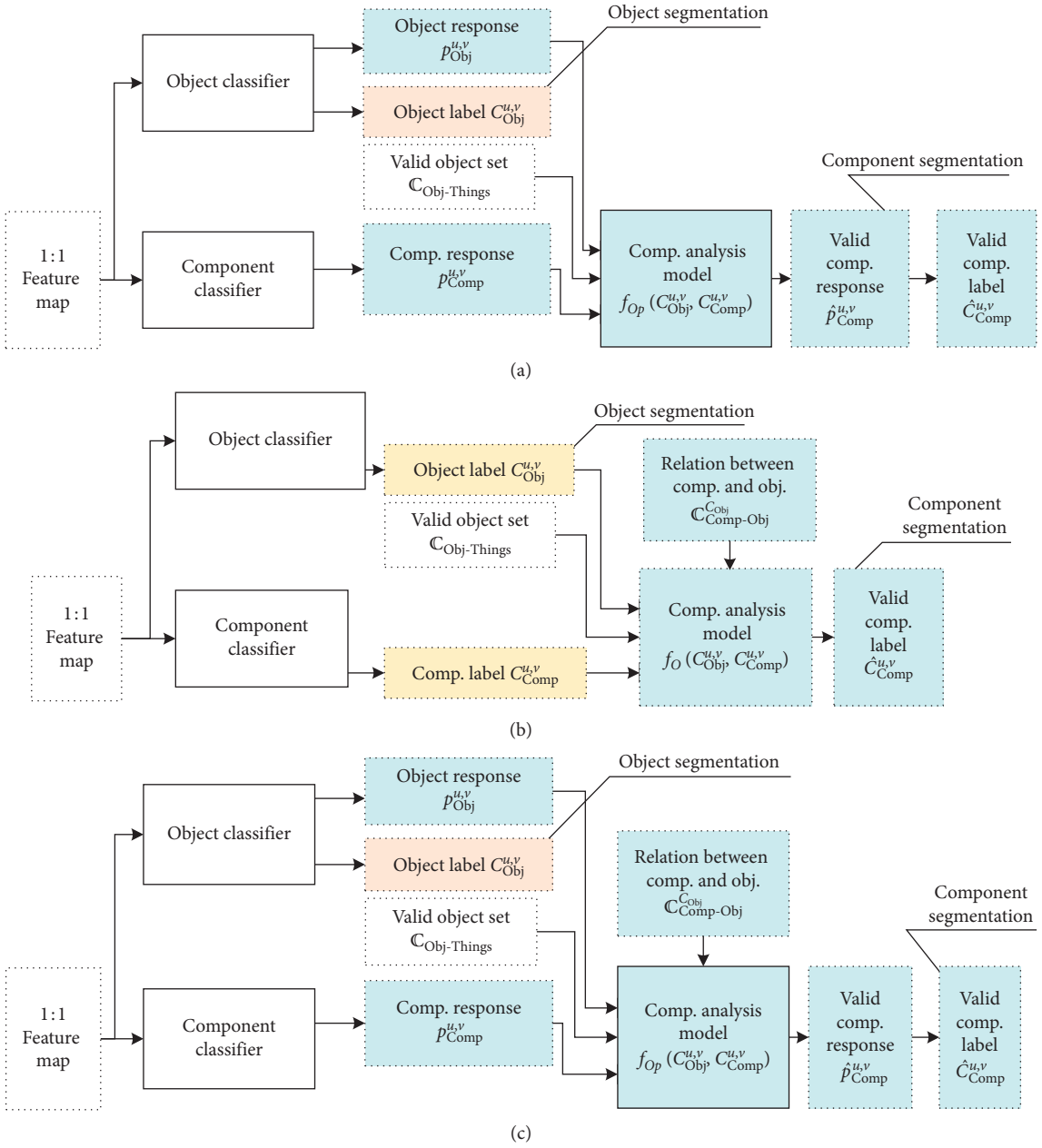


Figure 9: Optimized architecture with the component analysis module. (a) Replace $1 \times \text{argmax}_k p_{\text{Comp}-k}$ with $\text{argmax}_k \widehat{p}_{\text{Comp}-k}$ to optimize the module. (b) Analyze $C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{\text{Obj}}}_{\text{Comp}-\text{Obj}}$ to optimize the module. (c) Replace $1 \times \text{argmax}_k p_{\text{Comp}-k}$ with $\text{argmax}_k \widehat{p}_{\text{Comp}-k}$ and analyze $C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{\text{Obj}}}_{\text{Comp}-\text{Obj}}$ to optimize the module.

Table 2: Pixel accuracy and segmentation time of UPerNet with different component analysis modules (CAMs) on ADE20K component segmentation task.

	Backbone η_{main}	Backbone depth d_{main}	Decoder η_{decoder}	Comp. Analysis model	Comp. Segmentation accuracy $\text{PA}_{\widehat{\text{Part}}}$ (%)	Segmentation time T_{seg} (ms)
1	ResNet	50	PPM + FPN	—	48.30	483
2	ResNet	101	PPM + FPN	—	48.71	598
3	ResNet	152	PPM + FPN	—	48.89	721
4	ResNet	50	PPM + FPN + CAM	$1 \times \text{argmax}_k p_{\text{Comp}-k}$ [29]	53.62	490
5	ResNet	101	PPM + FPN + CAM	$1 \times \text{argmax}_k p_{\text{Comp}-k}$ [29]	53.96	604
6	ResNet	152	PPM + FPN + CAM	$1 \times \text{argmax}_k p_{\text{Comp}-k}$ [29]	54.18	726

Table 2: Continued.

	Backbone η_{main}	Backbone depth d_{main}	Decoder η_{decoder}	Comp. Analysis model	Comp. Segmentation accuracy $\text{PA}_{\widehat{\text{Part}}}$ (%)	Segmentation time T_{seg} (ms)
7	ResNet	50	PPM + FPN + CAM	$\text{argmax}_k \widehat{p}_{\text{Comp}-k}$	54.03	492
8	ResNet	50	PPM + FPN + CAM	$C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{Obj}}_{\text{Comp}-\text{Obj}}$	55.13	486
9	ResNet	50	PPM + FPN + CAM	$\text{argmax}_k \widehat{p}_{\text{Comp}-k} + C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{\text{Obj}}}_{\text{Comp}-\text{Obj}}$	55.62	496

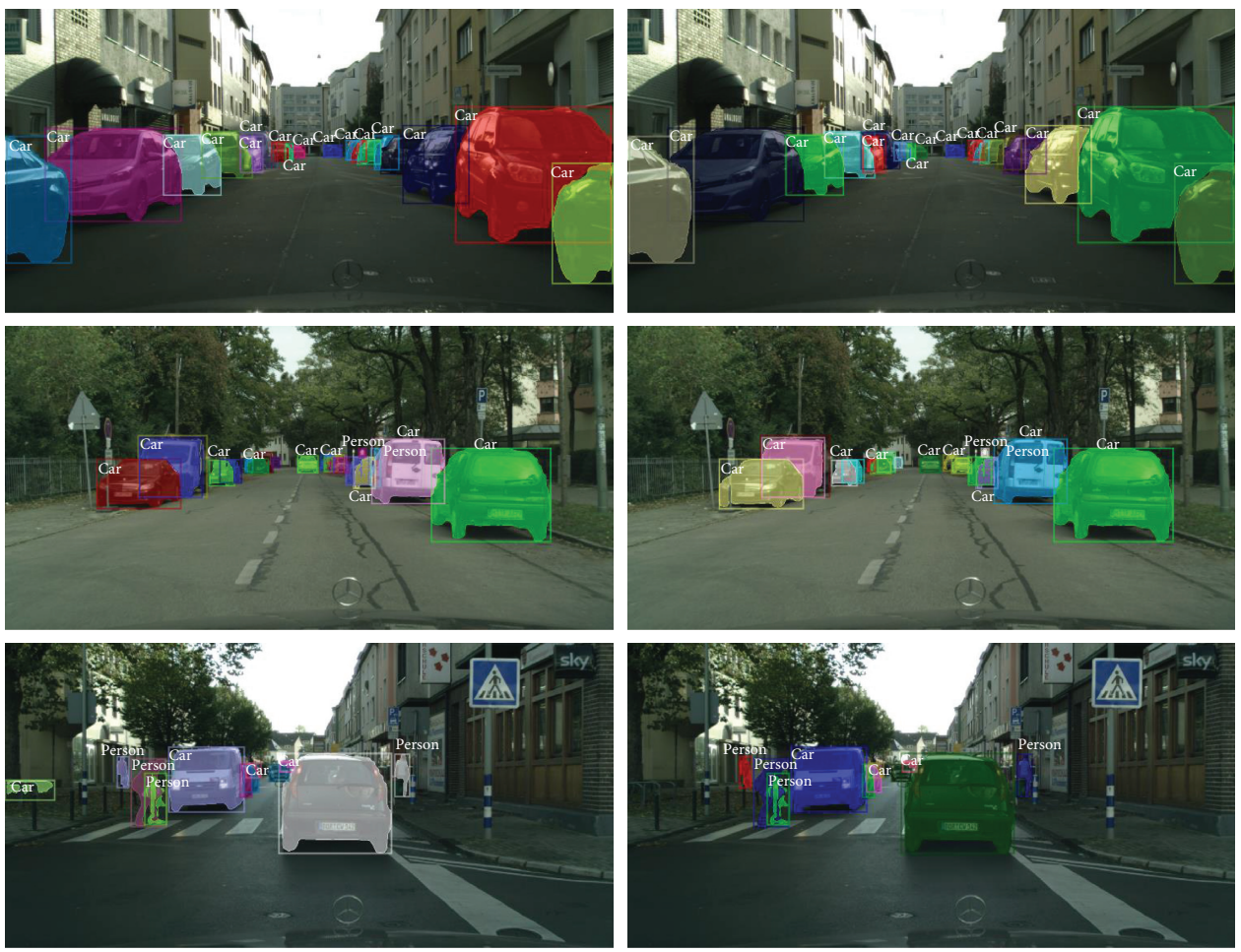
Table 3: Performances of different methods on CITYSCAPES instance-level semantic labeling task.

Method	AP (%)	$\text{AP}_{0.50}$ (%)	Segmentation time (ms)
SegNet	29.5	55.6	—
Mask R-CNN	32.0	58.1	—
UperNet	32.0	57.3	**447**
UperNet + CAM	**36.5**	**62.2**	451

CAM: Component Analysis Module.

Table 4: Mean average precision AP on class-level of the UPerNet with/without CAM in CITYSCAPES instance-level semantic labeling task.

Method	Person (%)	Rider (%)	Car (%)	Truck (%)	Bus (%)	Train (%)	Motorcycle (%)	Bicycle (%)
UperNet	36.0	28.8	51.6	30.0	38.7	27.3	**23.9**	19.4
UperNet + CAM	36.0	28.8	**53.0**	**34.3**	**57.0**	**37.5**	22.3	**23.8**



UPerNet UPerNet with the component analysis module

Figure 10: CITYSCAPES instance-level semantic labeling by UPerNet.

UPerNet

UPerNet with the component analysis module

Figure 11: Anticounterfeiting features detected by the UPerNet with/without the component analysis module.

Table 5: Pixel accuracy and segmentation time of UPerNet with different component analysis modules (CAM) for CNY anticounterfeit features via vision-based detection.

	Backbone η_{main}	Depth d_{main}	Decoder η_{decoder}	Component analysis module	$\text{PA}_{\widehat{\text{Part}}}$ (%)	$\text{AP}^{IoU_T=0.5}$(%)	T_{seg} (ms)
1	ResNet	50	PPM + FPN	—	88.50	85.3	483
2	ResNet	50	PPM + FPN + CAM	$1 \times \text{argmax}_k p_{\text{Comp}-k}$ [29]	90.38	96.1	490
3	ResNet	50	PPM + FPN + CAM	$\text{argmax}_k \widehat{p}_{\text{Comp}-k} + C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{\text{Obj}}}_{\text{Comp}-\text{Obj}}$	95.29	100	496

4. Conclusions

In this study, we performed semantic segmentation under a complex background using the encoder-decoder network to solve the issue of the mutually exclusive relationship between the semantic response value and the semantics of object/component in the semantic segmentation under a complex background for online machine vision detection. The following conclusions can be drawn from this study.

(i) Considering the mutually exclusive relationship between the semantic response value and the semantics of object/component, we selected the mathematical model of semantic segmentation under a complex background based on the encoder-decoder network for optimization. It was found that η_{main} = ResNet, d_{main} = 50 is the best encoder, and η_{decoder} = PPM + FPN is the best selected decoder.

(ii) We replaced $1 \times \text{argmax}_k p_{\text{Comp}-k}$ with $\text{argmax}_k \widehat{p}_{\text{Comp}-k}$. The component analysis module of $C^{u,v}_{\text{Comp}} \notin \mathbb{C}^{C_{\text{Obj}}}_{\text{Comp}-\text{Obj}}$ and UPerNet are considered to improve the performance of the encoder-decoder network.

(iii) The experimental results show that the component analysis module improves the performance of semantic segmentation under a complex background. Both $\text{PA}_{\widehat{\text{Part}}}$ and T_{seg} of the proposed model were better than those of the UPerNet with deeper d_{main}. Specifically, the accuracy improved from 48.89% to 55.62% and T_{seg} from 721 to 496 ms. By performing

vision-based detection with the 2019 CNY features, we showed that the proposed method improved $\mathrm{PA}^{\frown}_{\mathrm{Part}}$ from 90.38% to 95.29% while T_{seg} increased only slightly from 490 to 496 ms; $\mathrm{AP}^{I\delta U_T=0.1}$ also increased from 96.1% to 100%, detecting all the light transmission anticounterfeiting features without false detection, missing detection, or repeated detection.

The model in which $1 \times \mathrm{argmax}_k p_{\mathrm{Part}-k}$ was replaced with $\mathrm{argmax}_k \widehat{p}_{\mathrm{Part}-k}$ and the corresponding component analysis module improved the performance of the UPerNet encoder-decoder network. However, the efficiency improvement is affected by the accuracy of object segmentation. In our next study, we will investigate the applicability of machine learning to the component analysis module to achieve a higher performance in different applications.

Data Availability

The ADE20K Dataset used to support the findings of this study is available at http://groups.csail.mit.edu/vision/datasets/. The CITYSCAPES Dataset used to support the findings of this study is available at https://www.cityscapes-dataset.com. Its pretrained models and code are released at https://github.com/CSAILVision/semantic-segmentation325pytorch.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the Key-Area Research and Development Program of Guangdong Province (Grant no. 2019B010154003) and the Guangzhou Science and Technology Plan Project (Grant no. 201802030006).

References

[1] C. Szegedy, W. Liu, and Y. Jia, "Going deeper with convolutions," in *Proceedings of the Computer Vision and Pattern Recognition*, IEEE, Boston, MA, USA, pp. 1–9, June 2015.

[2] K. He, X. Zhang, and S. Ren, "Deep residual learning for image recognition," in *Proceedings of the Computer vision and pattern recognition*, IEEE, Las Vegas, NV, USA, pp. 770–778, June 2016.

[3] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, 2020.

[4] G. Liu, B. He, and S. Liu, "Chassis assembly detection and identification based on deep learning component instance segmentation," *Symmetry*, vol. 11, no. 8, 2019.

[5] R. Manish, A. Venkatesh, and S. Denis Ashok, "Machine vision based image processing techniques for surface finish and defect inspection in a grinding process," *Materials Today: Proceedings*, vol. 5, no. 5, pp. 12792–12802, 2018.

[6] L. Geng, Y. Wen, and F. Zhang, "Machine vision detection method for surface defects of automobile stamping parts," *American Scientific Research Journal for Engineering, Technology, and Sciences*, vol. 53, no. 1, pp. 128–144, 2019.

[7] M. M. Islam and J. Kim, "Vision-based autonomous crack detection of concrete structures using a fully convolutional encoder–decoder network," *Sensors*, vol. 19, no. 19, 2019.

[8] S. Zhou, D. Nie, E. Adeli, J. Yin, J. Lian, and D. Shen, "High-resolution encoder-decoder networks for low-contrast medical image segmentation," *IEEE Transactions on Image Processing*, vol. 29, pp. 461–475, 2020.

[9] S. Liu, J. Huang, and G. Liu, "Technology of multi-category legal currency identification under multi-light conditions based on AleNet," *China Measurement &Test*, vol. 45, no. 9, pp. 118–122, 2019, in Chinese.

[10] H. Kang and C. Chen, "Fruit detection and segmentation for apple harvesting using visual sensor in orchards," *Sensors*, vol. 19, no. 20, p. 4599, 2019.

[11] E. Pardo, J. M. T. Morgado, and N. Malpica, "Semantic segmentation of mFISH images using convolutional networks," *Cytometry Part A*, vol. 93, no. 6, pp. 620–627, 2018.

[12] G. Liu, S. Liu, and J. Wu, "Machine vision object detection algorithm based on deep learning and application in banknote detection," *China Measurement &Test*, vol. 45, no. 5, pp. 1–9, 2019, in Chinese.

[13] H. Gao, H. Yuan, and Z. Wang, "Pixel transposed convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 5, pp. 1218–1227, 2019.

[14] Q. D. Vu and J. T. Kwak, "A dense multi-path decoder for tissue segmentation in histopathology images," *Computer Methods and Programs in Biomedicine*, vol. 173, pp. 119–129, 2019.

[15] J. Huang and G. Liu, "The development of CNN-based semantic segmentation method," *Laser Journal*, vol. 40, no. 5, pp. 10–16, 2019, in Chinese.

[16] S. Nowozin, "Optimal decisions from probabilistic models: the intersection-over-union case," in *Proceedings of the Computer Vision and Pattern Recognition*, IEEE, Columbus, OH, USA, pp. 548–555, June 2014.

[17] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *Proceedings of the European Conference on Computer Vision*, pp. 340–353, IEEE, Florence, Italy, October 2012.

[18] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proceedings of the Computer Vision and Pattern Recognition*, IEEE, Boston, MA, USA, pp. 5353–5360, June 2015.

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

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

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," in *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, IEEE, Munich, Germany, October 2015.

[22] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *Proceedings of the European Conference on Computer Vision*, pp. 297–312, IEEE, Zürich, Switzerland, March 2014.

[23] N. D. Lane and P. Warden, "The deep (learning) transformation of mobile and embedded computing," *Computer*, vol. 51, no. 5, pp. 12–16, 2018.

[24] C. Qing, J. Ruan, X. Xu, J. Ren, and J. Zabalza, "Spatial-spectral classification of hyperspectral images: a deep learning framework with Markov Random fields based modelling," *Iet Image Processing*, vol. 13, no. 2, pp. 235–245, 2019.

[25] X. Li, Z. Liu, and P. Luo, "Not all pixels are equal: difficulty-aware semantic segmentation via deep layer cascade," in *Proceedings of the Computer Vision and Pattern Recognition*, IEEE, Honolulu, HI, USA, pp. 6459–6468, July 2017.

[26] B. Zhou, H. Zhao, X. Puig et al., "Semantic understanding of scenes through the ADE20K dataset," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, 2019.

[27] L.-C. Chen, "Rethinking atrous convolution for semantic image segmentation," 2017, https://arxiv.org/abs/1706.05587.

[28] H. Zhao, J. Shi, and X. Qi, "Pyramid scene parsing network," in *Proceedings of the Computer Vision and Pattern Recognition*, IEEE, Honolulu, HI, USA, pp. 6230–6239, July 2017.

[29] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proceedings of the European Conference on Computer Vision*, pp. 432–448, IEEE, Munich, Germany, September 2018.

[30] D. Kim, J. Kwon, and J. Kim, "Low-complexity online model selection with lyapunov control for reward maximization in stabilized real-time deep learning platforms," in *Proceedings of the Systems, Man and Cybernetics*, pp. 4363–4368, Miyazaki, Japan, January 2018.

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the Acm*, vol. 60, no. 6, pp. 84–90, 2017.

[32] M. Cordts, O. Mohamed, and S. Ramos, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Las Vegas, NV, USA, pp. 3213–3223, June 2016.