

Research Article

A Novel Approach to Optimizing Convolutional Neural Networks for Improved Digital Image Segmentation

Kongduo Xing,¹ Junhua Ku ,² and Jie Zhao²

¹College of Information Engineering, Hainan Vocational University of Science and Technology, Haikou, Hainan 571126, China

²School of Science, Qiongtai Normal University, Haikou, Hainan 571127, China

Correspondence should be addressed to Junhua Ku; junhuacoge@mail.qtnu.edu.cn

Received 19 August 2023; Revised 9 April 2024; Accepted 22 April 2024; Published 8 May 2024

Academic Editor: Tao Li

Copyright © 2024 Kongduo Xing et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To divide a digital image into individual parts that share similar characteristics is known as digital image segmentation, and it is a vital research subject in the field of computer vision. Object recognition, medical imaging, surveillance, and video processing are just a few of the many real-world contexts where this study could prove useful. While digital image segmentation research has come a long way, there are still certain obstacles to overcome. Segmentation algorithms frequently encounter challenges in achieving both accuracy and efficiency when confronted with intricate settings, noisy pictures, or fluctuating lighting conditions. The absence of established evaluation standards adds complexity to the process of performing equitable comparisons among different segmentation methodologies. Due to the subjective nature of photo segmentation, attaining consistent results among specialists can be challenging. The integration of machine learning and deep neural networks into segmentation algorithms has introduced new challenges, including the need for large amounts of annotated data and the interpretability of the outcomes. Given these challenges, the objective of this study is to enhance the segmentation model. To this end, this research suggests a model of convolutional neural networks that is optimal for digital picture segmentation. The model is based on a dense convolution neural network, and it incorporates a transfer learning technique to significantly boost the model's robustness and the quality of picture segmentation. The model's adaptability to new datasets is improved by the incorporation of a transfer learning method. As demonstrated by experimental results on two publicly available datasets, the suggested methodology considerably enhances the resilience of digital picture segmentation.

1. Introduction

Segmenting a digital image involves dividing it into several portions, each representing a different object or characteristic. Many computer vision and image processing applications use this method for object detection, shape recognition, and change detection. Segmentation techniques use intensity or color information to arrange similar images into distinct sections [1]. We must discover and isolate visual components having unique properties, such as edges, corners, textures, or colors and then isolate them. Depending on the objective and image complexity, thresholding, edge detection, clustering, and region expansion may be used. Binary segmentation labels pixels as foreground or background, gray-level segmentation divides the image into

regions, and color segmentation labels pixels by color. Segmenting images containing many objects, occlusions, or background noise is tough. To improve segmentation accuracy and resilience, researchers have used machine learning, prior knowledge, and many segmentation methods. Segmentation is used in medical imaging to detect tumors and other abnormalities, surveillance and security to detect items or persons in video streams, and robotics and automation to guide object movement or manipulation.

Digital image segmentation is a crucial process for various applications such as medical imaging [2], object recognition [3], autonomous driving [4], and video surveillance [5]. In the field of medical image analysis, Li et al. [6] delve into automatic liver segmentation using deep learning, reviewing various models, including CNNs and

GANs, along with their pros, cons, and applications in liver segmentation. The authors extensively explore deep learning-based autonomous liver segmentation methods, weighing their merits and drawbacks. Wang et al. review deep learning-based semantic segmentation for medical picture analysis [7]. Dalvand et al. proposed a parallel fusion method using majority voting technology to enhance the credibility and user interaction of interactive image segmentation algorithms and introduced a spiking neural-like P system model to reduce the computational burden [8]. Narayan et al. summarized the importance and use of image segmentation in the field of medical images. At the same time, they also analyzed the problems, current technology, and mathematical implementation in medical imaging [9]. Shukla et al. proposed a new method called TrustMIS to solve this problem. TrustMIS improves performance and trustworthiness by studying trustworthiness and selecting the most trustworthy models [10]. Most of the above studies introduce deep learning methods to improve image segmentation performance. However, other problems will be introduced in the process of improving segmentation accuracy.

Recent years have seen significant advances in digital picture segmentation. Deep learning algorithms such as CNNs excel in semantic segmentation, instance segmentation, and object detection. Unsupervised and weakly supervised learning methods have also shown potential in photo segmentation. Despite the progress in digital image segmentation, several challenges remain. When applied to new data, segmentation models may lack the robustness and generalization needed to handle complicated problems. In order to solve the above problems, this paper has carried out a series of work. These are the primary contributions of this study: (1) To enhance the resilience of the image segmentation model, we propose the implementation of a convolutional neural network model based on optimization. Our primary objective in this model design is to enhance the robustness of dense convolutional neural networks by refining their foundation across various scenarios and datasets. (2) Implementation of a mechanism for transfer learning. To further enhance the model's robustness and the accuracy of image segmentation, we deftly incorporated the transfer learning mechanism into the architecture. By utilizing this mechanism, the model is capable of more effectively adjusting to diverse datasets, thus significantly augmenting its adaptability in real-world scenarios. (3) A dense convolutional neural network is chosen as the foundational architecture. Leveraging the dense convolutional neural network architecture, it is capable of capturing image features and contextual information more effectively, thereby enhancing its capability to assist with image segmentation tasks. Experimental demonstration: We conduct exhaustive experiments on two public datasets, and the results show that the proposed model achieves significant performance improvements in digital image segmentation. Specifically, we observe that the robustness of the model is significantly

improved, while the segmentation accuracy is also effectively improved.

2. Application of Deep Learning in Image Segmentation

2.1. The Principle of Deep Learning Applied to Digital Image Segmentation. The process of splitting an image into several segments or areas, each of which corresponds to a separate object or component of an object, is referred to as image segmentation. This process takes place during image processing. Because it can automatically learn relevant characteristics and patterns in images, deep learning is a technique that is becoming increasingly popular for application in image segmentation.

When it comes to image segmentation, the fundamental idea behind deep learning is to train a convolutional neural network (CNN) to learn a mapping from the input image to a pixel-by-pixel segmentation mask. This is how deep learning works. The CNN is made up of numerous layers, each of which is responsible for learning progressively complicated properties from the input image. The final layer generates a segmentation mask that assigns a predicted class label (such as "object" or "background") to each pixel in the input image. This label may indicate whether the pixel represents an object or the backdrop. After that, the output segmentation mask may be put to use to either isolate the object of interest or conduct additional research. The basic idea behind using a deep learning algorithm to the segmentation of digital images is illustrated in Figure 1.

While deep learning has shown promise in the field of picture segmentation, it still faces a number of obstacles. Large amounts of labeled data are required to train the model, which presents a challenge. Annotation is the labor-intensive and resource-intensive process of manually assigning a class label to each pixel in an image. Not only that but the training data's high quality and quantity determine the segmentation's final quality. One difficulty is that the data may have an unbalanced representation of classes, with some classes being more common than others. This can cause segmentation performance for the minority class to suffer because of bias toward the majority class. Data augmentation, loss function balancing, and ensemble learning are only few of the methods offered to deal with class imbalance. As an added complication, deep learning models are notoriously complex and computationally costly, making them challenging to train and roll out. Model performance is highly sensitive to architectural, hyperparameter, and optimization algorithm selections. Overfitting, in which the model memorizes the training data instead of learning to generalize to new data, is another prominent issue in deep learning applied to picture segmentation. Although deep learning has demonstrated impressive promise for picture segmentation tasks, it still takes significant thought and expertise in data annotation and model design to provide accurate results.

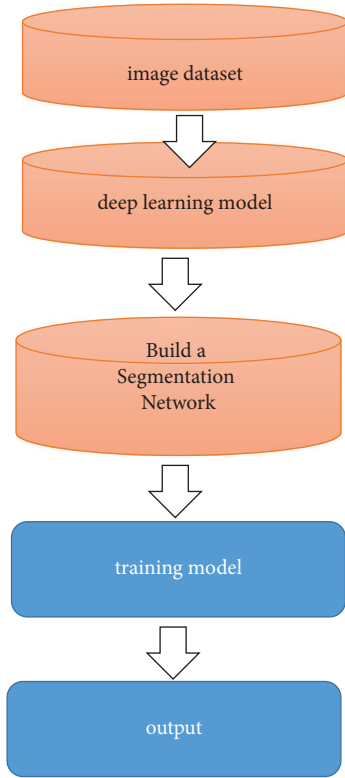


FIGURE 1: Principle of image segmentation based on deep learning.

2.2. Convolution Neural Network

2.2.1. Basic Network Organization. Computer scientists have developed the CNN algorithm by assimilating pertinent information from various fields. Researchers in the past abstracted the transmission principle and procedure from studies of animal visual cortex signals to the computer in order to create CNN [11]. CNNs mimic the brain's signal transmission mechanism at each stage of the network, including the pooling layer and the convolution layer. Figure 2 depicts the modular nature of such a building. Many neurons make up the various "layers" of a network.

Layer one is the input. Preprocessed images of food are used as the input for a convolutional neural network for use in pattern identification. The overall model's detection accuracy in a convolutional neural network depends critically on the image quality input. Input image data balance must be maintained. Every food picture provided in the input layer is a matrix that needs to be transformed into a single dimension. The images have three channels because they are RGB color images. Rotating, scaling, and slicing the food sample can increase the sample size and decrease the overfitting problem that occurs when training a model, all of which are important to guarantee the diversity of data and to enlarge the sample. To further reduce visual interference on convolutional neural networks, some approaches exist for preprocessing numerous images, such as averaging, normalization, and other operations.

Layer two is the convolutional filter. The convolution layer is the backbone of the entire CNN. A model of neuronal behavior in the human brain has been used to inform

the design of this layer. In the architecture of the convolutional layer, it comprises multiple convolutional strata. The connectivity paradigm within this structure is such that nodes at different levels do not exhibit direct interconnections. However, nodes residing in adjacent convolutional layers are linked through a transmission relationship, allowing selective data intake from the preceding layer's nodes. This organizational schema ensures comprehensive representation of the entire image's information by assigning distinct nodes to encapsulate various facets of the image. Consequently, individual nodes are not burdened with the necessity of processing the entirety of the image.

Figure 3 elucidates the process inherent in the convolution operation. This process utilizes convolutional computations to synthesize multiple features. It achieves this by amalgamating data from numerous nodes in the antecedent layer into a singular node in the subsequent layer. The dimensions of the convolution matrix are congruent with the size of the convolution kernel. Furthermore, the values constituting the convolution kernel are adjustable, allowing for contextual customization to optimize feature extraction.

The activation function is a vital part of the convolution layer. Linearity holds true even if all convolutions are linked through convolution calculation. As time went on, though, experts suggested that an activation function may transform a linear structure into a nonlinear one. Activation functions, which are what connect the convolutions, are more in line with the way true recognition works. The weighted sum process is generated after the image has been processed in the input layer, operated on in the convolution layer, and so on. Here is how it works,

$$X'_j = f \left(\sum_{i \in M} x_i^{l-1} * k_{ij}^l + b_j^l \right), \quad (1)$$

where X'_j represents the j th feature of the l th volume layer, x_i^{l-1} represents one of the features, and b_j^l indicates the offset of the feature.

(1) *Pool Layer.* In order to join two volumes together, a pooling layer is necessary. The pooling layer is there to avoid any overfitting. It operates in a manner analogous to that of a convolution layer. You can choose your pooling matrix in the pooling layer and then use the window sliding to get at the next layer's important data. There are two primary approaches to pooling in the pooling layer: maximal pooling and average pooling. With maximum pooling, the largest possible values are retained in the output pool. When you pool an area's average value, you are doing average pooling. Two pooling techniques, with a convolution core size of 2×2 , are shown in the process and output in Figure 4. The increment is 2 steps.

The pooling layer must also have some of the aforementioned traits, but the most crucial one is that it must preserve the original image's attributes. We can guarantee that the picture's characteristics will not change no matter what we do to the photo. This method is useful for image

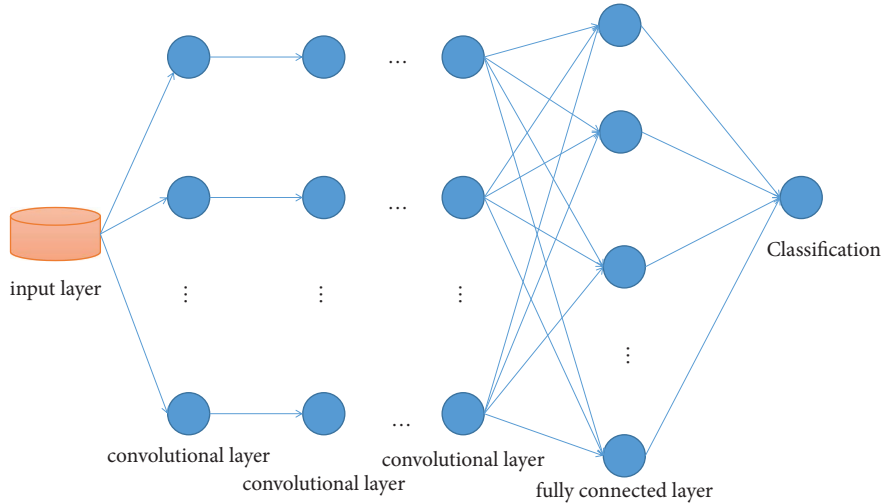


FIGURE 2: Network structure of CNN.

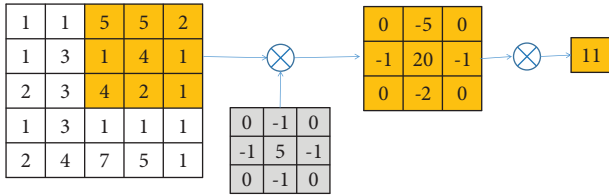


FIGURE 3: Convolution operation.

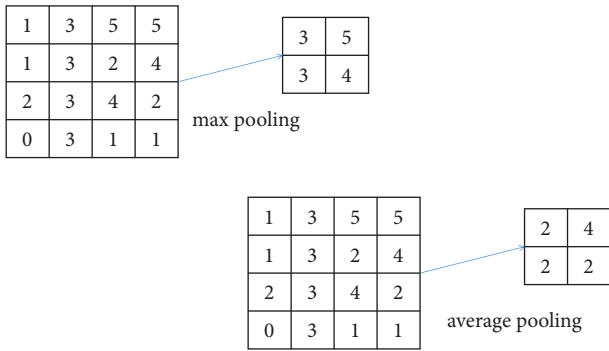


FIGURE 4: Pooling operation.

recognition since it helps preserve image characteristics while also reducing data duplication.

There is a complete layer of connections in place in the fourth position. CNN’s final layer is the entire connection layer. After the convolution is calculated, the image is sent on to the entire connection layer. The term “full connection layer” is used to describe the process of connecting numerous independent networks into one larger one. Often times, the connection layer is multilayered. Several full connection levels can be configured to meet varying requirements. The whole connection layer can be expressed as

$$x^i = f(W^i x^{i-1} + b^i), \quad (2)$$

where x_i^{i-1} represents the characteristics of the $i-1$ layer in the network and W^i and b^i represent the weights and offsets, respectively.

CNNs can use one of four standard activation functions—the sigmoid, Tanh, ReLU, or Leak-ReLU. Here are the written forms for the four activation functions,

$$g(x) = \frac{1}{1 + e^{-x}}, \quad (3)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}, \quad (4)$$

$$f(x) = \begin{cases} x, & x > 0, \\ 0, & x \leq 0. \end{cases} \quad (5)$$

$$f(x) = \begin{cases} x, & x > 0, \\ ax, & x \leq 0. \end{cases} \quad (6)$$

2.2.2. Communication Mode of Network. With a convolution neural network, information can travel both forward and backward in the network [12]. CNNs are trained in image recognition processing utilizing forward propagation and backward propagation techniques, respectively, until a satisfactory model is achieved. The primary procedure can be stated by defining the input picture as I and the output image as O using convolutional neural networks. In order to train, O must be fine-tuned over and over again until the desired output is achieved. The act of communicating is a continual process of learning and training.

The output layer of a CNN is responsible for deriving the final output of the network’s processing of the input signal. We refer to the mode of propagation used during this phase of training as forward propagation, and its formula is as follows:

$$a'_j = f\left(\sum_k w'_{jk} a_k^{l-1} + b'_j\right), \quad (7)$$

where a'_j represents the output node of the j th neuron in the l -layer network, a_k^{l-1} represents the output value of the k -th neuron in the layer $l-1$ network, and \mathbf{W} represents the weight of neurons. z is the activation function, and its expression is as follows:

$$\begin{cases} z' = w' a^{l-1} + b', \\ a' = f(z'). \end{cases} \quad (8)$$

The reverse of the forward propagation mode is known as the backward propagation mode. The basic goal of backward propagation in a convolution neural network is to minimize the error problem. The heart of back-propagation is a way for continuously adjusting the trend by using the loss function to tweak the weight and bias. If you are familiar with the forward propagation approach of ideology, you will see a big difference between this and that.

Assuming x is the value used as the sample from the image, with n being the total number of samples, the output is given by the formula $y = y(x)$. The formula for its back-propagation is as follows:

$$C = \frac{1}{2n} \sum_x \|y(x) - a(x)'\|^2. \quad (9)$$

where e is the expected value.

3. TLDenseNet Model

3.1. Image Segmentation Based on TLDenseNet Model. The use cases for digital picture segmentation are numerous, as are the types of images that can be segmented. In this research, we propose an improved version of DenseNet [13] (TLDenseNet)—one of the best CNN models—by integrating the transfer learning mechanism to better handle pictures with varying attributes and segmentation needs. Figure 5 depicts the basic idea behind this model-based approach to digital picture segmentation.

First, the dataset is entered into the initial dense convolution network to obtain the dense network with parameter information, as shown in the figure. Using this network as a foundation, we offer a transfer learning technique to build a high-density convolutional network with a built-in transfer learning function. Second, the aforementioned network is trained using the image to be segmented as input, and a model for doing so is obtained. Lastly, the test picture can be segmented using the trained model to achieve the final segmentation result.

3.2. DenseNet. As the dense convolution network does not have to retrain and learn the redundant feature map, it may function with fewer parameters than the standard convolution network. In this network architecture, new information is very distinct from existing information that is kept [14]. The DenseNet architectural framework is characterized by its slender convolution layers, a direct

consequence of the profuse interconnectivity among nodes within the network. Augmentations to this network are constituted by a modest assemblage of feature maps, which notably retain their original, unaltered state. The ultimate classification process within this architecture is predicated upon an integrative analysis of all the feature maps present across the network, leveraging their collective information for accurate prediction outcomes. When compared to the residual network, the dense connection technique proposed by DenseNet is more radical. As can be seen in Figure 6, DenseNet's dense connection mechanism consists of blocks of connections. It is clear how the layers are interconnected and how the feature maps of successive levels are identical in size and used as input by the one before them. The first layer takes its input from a merged $[x_0, x_1, \dots, x_{l-1}]$ feature diagram from all lower layers. Layer l 's method of calculation is as follows:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]), \quad (10)$$

where $[x_0, x_1, \dots, x_{l-1}]$ is the splicing of all previous feature maps and H_l represents a composite operation composed of batch normalization, corrected linear unit, and convolution.

A deep convolutional network with L layers has a total of $L(L+1)/2$ connections. This is a highly interconnected node, especially when compared to the residual network. DenseNet also directly merges feature maps from several layers. This process enhances efficiency since it allows for enhanced feature reuse. In this regard, DenseNet differs significantly from the residual network.

The hyperparameter k is called the growth rate of the network. The growth rate k represents the number of characteristic graphs generated by the function H of each layer, so the calculation formula of the number of characteristic graphs k of the first layer as input is

$$k_l = k_0 + (l-1)k. \quad (11)$$

For example, let k_0 stand for the total number of characteristic graphs that were used as input. DenseNet is distinctive from the current network architecture in that it is much more compact. A more modest rate of expansion can nonetheless yield satisfactory outcomes. This is because the network acquires "collective knowledge" at the dense connection block, where feature maps from each successive layer can be coupled. Instead of having to repeat between layers, as in a conventional network architecture, calling the characteristic graph is possible at any point in the network.

Network blocks with dense connections and a transition layer make up the bulk of DenseNet's architecture. DenseNet architecture is depicted in Figure 7. There are three clusters of connections representing significant parts of the structure in the diagram. The dense convolution network can be segmented into many dense connection blocks to ensure that the feature graph sizes are uniform across the network and that splicing will not be an issue.

Within the DenseNet architecture, the dense connection block incorporates a bottleneck layer, designed to reduce computational demands. This reduction is achieved

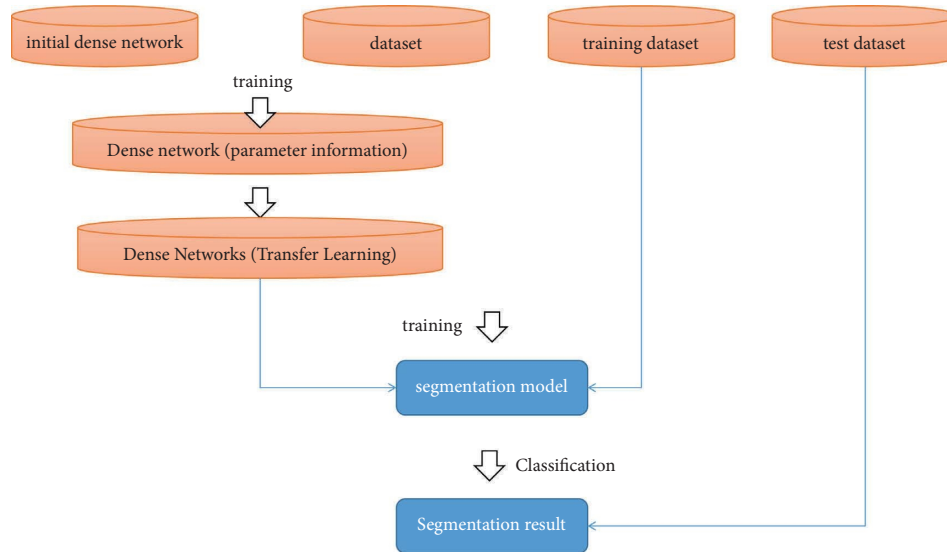


FIGURE 5: Image segmentation principle based on TLDenseNet model.

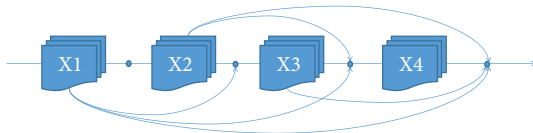


FIGURE 6: Dense connection block.

primarily through the integration of 1×1 convolutions. The structure of this layer follows a sequence: batch normalization, followed by a modified linear unit, a 1×1 convolution, another batch normalization, a second modified linear unit, and culminating with a 3×3 convolution. This configuration is referred to as the DenseNet-B structure. The primary function of the 1×1 convolution within this sequence is to diminish the number of features, thereby enhancing computational efficiency. Additionally, this convolution facilitates the amalgamation of features across various channels.

Adjacent to each dense connection block is a transition layer, tasked with reducing the number of feature maps. This reduction is accomplished through a combination of convolution and pooling processes. The transition layer is composed of three distinct components: a BN (batch normalization) layer, a convolution layer, and a pooling layer. If a dense connection block outputs “m” characteristic maps, the ensuing transition layer will produce “A” characteristic maps as its output. Here, “B” symbolizes the compression coefficient, denoting the proportion of feature map reduction. This aspect of the DenseNet architecture is termed DenseNet-C.

Both the bottleneck and transition layers are integral to a model variant known as DenseNet-BC. The transition layer’s role is pivotal; it compresses and consequently reduces the channel count before transmitting data to the subsequent dense connection block. This function underscores the essentiality of the transition layer within the DenseNet framework.

3.3. Transfer Learning Mechanism. Training data are required by traditional classification algorithms; however, the labeling cost of these training data is very high, and the amount of training data must be continuously grown to keep up with the demand. The vast amount of previously labeled data is lost when we have to relabel the training set in response to a similar demand. What insights can we draw from these numbers to better prepare us for the work ahead of us? Transfer learning aims to address this issue [15]. Instead of being a single algorithm model, the term “transfer learning” refers to a set of approaches that have proven effective in the modern study of deep learning. We apply the model’s learnings from one set of problems to the solution of another set of problems that are conceptually similar. As a result, the model can use its prior knowledge to speed up its training on the current task. Figure 8 depicts the specific principle in detail.

We may essentially classify the transfer learning strategy into three types: (1) The sample-based transfer learning approach selects data from the source domain dataset that has a probability distribution that is close to that of the target domain data. (2) A transfer learning strategy based on features. We reduce the dissimilarity between source and target domain samples by projecting them into renewable and hilt space (RKHS). (3) A parameter/model-based transfer learning approach. To use the network as a feature extractor before training, just swap out the last layer for a classifier. During fine-tuning, all weights are recalculated using the initialized model parameters from before training. It is not advised to increase the learning rate beyond the learning rate during the tuning procedure. A better option is $1e-5$ most of the time.

4. Experimental Structure and Analysis

4.1. Experimental Data. Several picture segmentation datasets are available. Currently, the most researched datasets are those for medical image segmentation and scene

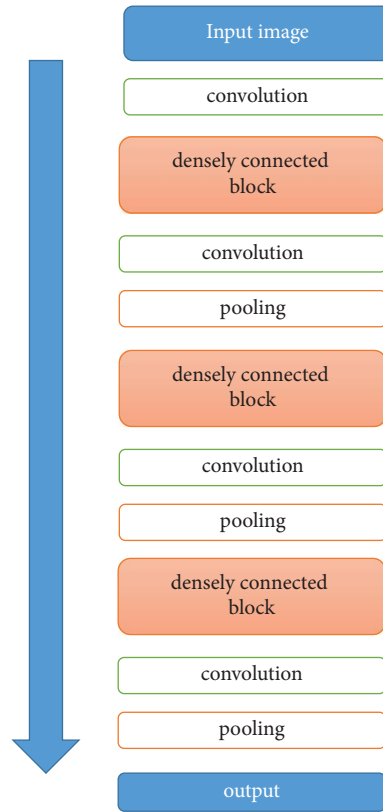


FIGURE 7: DenseNet structure diagram.

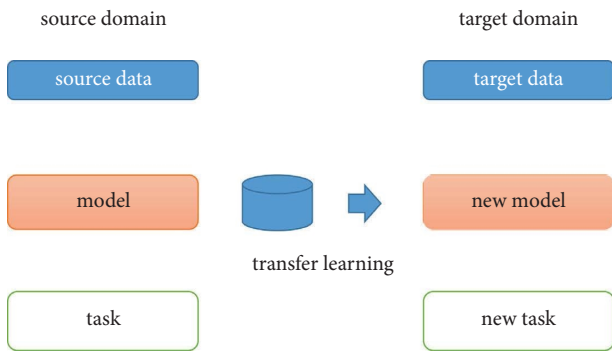


FIGURE 8: Transfer learning mechanism in deep learning.

analysis. Because of this, we chose to conduct our experiment using the liver image segmentation dataset [16] and the MIT scene analysis dataset [17].

The 2017 IEEE International Biomedical Imaging Symposium (ISBI) Liver Tumor Segmentation Challenge (LiTS) dataset is used in this paper. In addition, the LiTS dataset includes 400 individual sheets, each of which is a 512 by 512 CT image of a patient with hepatocellular carcinoma. In addition, the sample data are heterogeneous due to the vast variances across each CT, which successfully emphasizes the generalizability of model segmentation outcomes.

The MIT scenario analysis dataset can be found at <https://sceneparsing.csail.mit.edu/> for download. Scene analysis may be practiced and tested on a consistent, industry-standard platform, thanks to this data collection.

There are more than 20,000 high-quality photos of scenes, each annotated with precise details about the objects in the shot. The dataset is broken down into a training set of 20,000 photographs, a verification set of 2,000 images, and a separate set of test data. There are 150 different types of items and scenes represented in the dataset, from roads and grass to the sky and people. Rather than appearing randomly, as would be the case in real life, the things in the image have the appearance they would have in a realistic computer-generated environment. An image's semantic segmentation mask is generated by the segmentation algorithm's predictions about which semantic category each pixel belongs to.

4.2. Experimental Environment and Settings. The experimental test bed utilized in this study is configured with a Windows 10 64 bit operating system and is powered by an Intel Core i7-9700K CPU. It also includes a GTX 1080Ti GPU and is equipped with 16 GB of RAM. The software development environment is comprised of *Python* 3.6 and version 1.1.0 of an unspecified software component. For the training of the model, specific parameters are employed. These include a training duration of 100 epochs and a learning rate set at $1e-2$. Additionally, the model processes data in batches, with each batch containing 64 instances. Pixel Accuracy (PA) and Intersection-Over-Union (IoU) are the two major performance indicators that are utilized in relation to the evaluation of the effectiveness of the segmentation algorithm [18]. PA is defined as the

TABLE 1: Segmentation results of liver tumor dataset.

Index\Model		CNN	RNN	LSTM	Unet	DenseNet	TLDenseNet
PA	Mean	0.9153	0.9024	0.9341	0.9445	0.9376	0.9626
	std	0.0465	0.0643	0.0245	0.0172	0.0354	0.0115
IoU	Mean	0.8799	0.8645	0.9006	0.9342	0.9101	0.9452
	std	0.0097	0.0175	0.0202	0.0123	0.0087	0.0106

TABLE 2: Split results of MIT dataset.

Index\Model		CNN	RNN	LSTM	Unet	DenseNet	TLDenseNet
PA	Mean	0.6037	0.6632	0.7017	0.6452	0.7123	0.7328
	std	0.0242	0.0193	0.0146	0.0087	0.0901	0.0075
IoU	Mean	0.3253	0.3735	0.3921	0.3590	0.4017	0.4205
	std	0.0194	0.0215	0.0176	0.0131	0.0201	0.0102

proportion of pixels in a picture that are correctly identified. In essence, PA is a measurement of the ratio of pixels that have been correctly labeled to the total number of pixels in the image. The Intersection-Over-Union metric, on the other hand, is a measure that provides an indication of the precision of the segmentation algorithm. It does this by providing a measurement of the overlap that exists between the predicted segmentation and the ground truth,

$$PA = \frac{\sum_{i=0}^n P_{ii}}{\sum_{i=0}^n \sum_{j=0}^n P_{ij}} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (12)$$

when there are n categories, plus the extra one for the context; the total number of categories is $n + 1$. The number of predicted category I pixels is denoted by p_{ii} , where I is the category of the actual pixels. The number of pixels whose actual category is I that are incorrectly identified as category j is denoted as p_{ij} , where p_i is the total number of pixels and j is the anticipated category. A true positive (TP) is a case where both the label and the anticipated value are correct. As the name implies, TN is the total number of instances where both the label and the forecasted value were incorrect. When the label is negative but the predicted value is positive, we say that there was a certain amount of false positives. The false negative (FN) is a positive label and a negative prediction. Four added together equals the full complement of picture elements. Correct categorization counts as TP + TN.

In semantic segmentation, Intersection-Over-Union (IoU) is a popular indicator. The overlap region between the prediction segmentation and the labels is the IOU. This indication goes from 0 to 1, with 0 denoting fully separate regions and 1 indicating full overlap between them. The following is the mathematical formula:

$$IoU = \frac{|A \cap B|}{|A \cup B|} = \frac{TP}{TP + FP + FN}. \quad (13)$$

4.3. Analysis of Experimental Results. In this study, we compare the proposed model to several others used for digital image segmentation, including CNN [19], recurrent neural network (RNN) [20], long short-term memory

(LSTM) [21], Unet [22], and DenseNet [23]. Tables 1 and 2 display the models' segmentation results on the MIT liver tumor dataset. The experimental data are the average value obtained after training the model five times.

Table 1 shows the segmentation results of six deep learning models for liver dataset. First, observing the PA indicator, TLDenseNet performed the best, reaching 0.9626, which has higher pixel-level accuracy than other models. Followed by Unet and DenseNet, with 0.9445 and 0.9376, respectively, their performance is also relatively good. The PA means of RNN and LSTM are 0.9024 and 0.9341, respectively, which are slightly lower than other models. This shows that TLDenseNet has significant advantages in pixel-level classification accuracy. Second, when examining the IoU index, TLDenseNet also performed well, reaching 0.9452, showing its excellent performance in terms of overlap of target areas. The IoU means of Unet and DenseNet are 0.9342 and 0.9101, respectively, which are still at a relatively high level. The IoU means of RNN and LSTM are 0.8645 and 0.9006, respectively, indicating that their overlap in the target area is relatively low. In addition, we also observed the standard deviation (std) of each model on the two indicators. The standard deviations of TLDenseNet on PA and IoU are relatively small, 0.0215 and 0.0156, respectively, showing the relative stability of its model performance. The standard deviations of other models are also within the acceptable range, but TLDenseNet performs better in this regard.

Table 2 shows the segmentation results of six deep learning models for MIT datasets. First, as can be seen from the PA indicator, TLDenseNet performs best at the level of 0.7328, which has stronger performance in pixel-level accuracy than other models, followed by DenseNet and Unet, with 0.7123 and 0.6452, respectively. The PA means of RNN and LSTM are 0.6632 and 0.7017, respectively, showing their relatively low performance in this regard. Second, it can be seen from the IoU indicator that TLDenseNet reaches the highest value at the level of 0.4205, showing superior performance in terms of overlap of target areas. The IoU means of DenseNet and Unet are 0.4017 and 0.3590, respectively, which are also at a relatively high level. The IoU means of RNN and LSTM are 0.3735 and 0.3921, which are relatively

low. In terms of standard deviation, the standard deviation of TLDenseNet on PA and IoU is relatively small, showing that its performance on the MIT dataset is relatively stable. The standard deviation of DenseNet is large, especially on PA, which indicates that its performance on this dataset fluctuates greatly.

5. Conclusion

Segmenting a digital image means breaking it up into smaller pieces that each reflects a different feature of the image. Segmenting an image digitally is difficult since it calls for the algorithm to precisely locate the edges of objects and distinguish them from the backdrop. Since deep learning algorithms can automatically discover useful features from large volumes of data, they have become valuable tools for digital image segmentation. CNNs are a common choice for digital picture segmentation problems since they are a sort of deep learning algorithm. CNNs may be trained to recognize important picture features and then utilize that knowledge to assign a label to each pixel in an image. Digital picture segmentation performance has been greatly enhanced by recent advancements in deep learning algorithms. This paper enhances the convolutional neural network's robustness and segmentation performance by introducing a transfer learning method. The model described in this research exhibits strong stability and segmentation accuracy in both medical picture segmentation and general image segmentation tasks. There are still obstacles to be overcome, notwithstanding the achievements of deep learning algorithms in digital picture segmentation. Large amounts of labeled data are required to train the algorithms, which is a significant problem. To generate huge datasets for various applications, image annotation is a time-consuming and costly operation. Future research can focus on developing unsupervised or weakly supervised image segmentation methods. These methods can reduce the need for large amounts of labeled data, thereby reducing the cost and complexity of training algorithms.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The research presented in this article was guided academically by the affiliated institution without financial support, and neither the submitter nor other collaborators had any conflicts of interest with the institution or any other parties.

Acknowledgments

This work was supported by the Hainan Provincial Natural Science Foundation of China under Grant No. 621RC599. The paper is also partly supported by the Hainan Provincial Free Trade Port Key Laboratory of Shipping Economic Development and Property Rights Digitalization.

References

- [1] J. Smith, M. Johnson, and A. Garcia, "New advances in imaging techniques," *European Journal of Radiology*, vol. 135, no. 2, pp. 178–187, 2021.
- [2] S. Kim, J. Park, and K. Lee, "Recent advances in ultrasonography for breast cancer," *European Radiology*, vol. 31, no. 5, pp. 2789–2799, 2021.
- [3] 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.
- [4] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2018.
- [5] Q. Zeng, L. Feng, and Y. Zhao, "LPNet: license plate recognition via deep neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1163–1173, 2020.
- [6] Z. Li, X. Liang, Y. Li, and J. Yao, "Deep learning-based automatic liver segmentation: a survey," *Journal of Healthcare Engineering*, vol. 2019, pp. 1–17, 2019.
- [7] G. Wang, X. Zhang, Y. Wang, Y. Chen, and J. Tian, "Deep learning-based semantic segmentation for medical image analysis: a survey," *Neurocomputing*, no. 396, pp. 410–427, 2020.
- [8] M. Dalvand, A. Fathi, and A. Kamran, "Spiking Neural P System with weight model of majority voting technique for reliable interactive image segmentation," *Neural Computing & Applications*, vol. 35, no. 12, pp. 9035–9051, 2022.
- [9] V. Narayan, M. Faiz, P. K. Mall, and S. Srivastava, "A comprehensive review of various approach for medical image segmentation and disease prediction," *Wireless Personal Communications*, vol. 132, no. 3, pp. 1819–1848, 2023.
- [10] S. Shukla, L. Birla, A. K. Gupta, and P. Gupta, "Trustworthy medical image segmentation with improved performance for in-distribution samples," *Neural Networks*, vol. 166, pp. 127–136, 2023.
- [11] Y. LeCun, "Generalization and network design strategies," *Connectionism in perspective*, vol. 19, no. 18, pp. 143–155, 1989.
- [12] K. Pettersson, A. Karzhou, and I. Pettersson, "A feedforward neural network for modeling of average pressure frequency response," *Acoustics Australia*, vol. 50, no. 2, pp. 185–201, 2022.
- [13] A. Ayad and M. E. Abdulmunim, "Detecting abnormal driving behavior using modified DenseNet," *Iraqi Journal for Computer Science and Mathematics*, vol. 4, no. 3, pp. 48–65, 2023.
- [14] Z. Y. Wang and S. W. Ji, "Smoothed dilated convolutions for improved dense prediction," *Data Mining and Knowledge Discovery*, vol. 35, no. 4, pp. 1470–1496, 2021.
- [15] K. Fernandes and J. S. Cardoso, "Hypothesis transfer learning based on structural model similarity," *Neural Computing & Applications*, vol. 31, no. 8, pp. 3417–3430, 2019.
- [16] G. K. Mourya, S. Paul, A. Handique, U. Baid, P. V. Dutande, and S. N. Talbar, "Modified U-Net for fully automatic liver segmentation from abdominal CT-image," *International Journal of Biomedical Engineering and Technology*, vol. 40, no. 1, pp. 1–17, 2022.

- [17] 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.
- [18] J. D. Bodapati, R.K. Sajja, and V. Naralasetti, "An efficient approach for semantic segmentation of salt domes in seismic images using improved UNET architecture," *Journal- The Institution of Engineers: Serie Bibliographique*, vol. 104, no. 3, pp. 569–578, 2023.
- [19] D. Im, D. Han, S. Choi, S. Kang, and H. J. Yoo, "DT-CNN: an energy-efficient dilated and transposed convolutional neural network processor for region of interest based image segmentation," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I-REGULAR PAPERS*, vol. 67, no. 10, pp. 3471–3483, 2020.
- [20] T. Le, G. Bui, and Y. Duan, "A multi-view recurrent neural network for 3D mesh segmentation," *Computers & Graphics*, vol. 66, pp. 103–112, 2017.
- [21] J. Donahue, L. A. Hendricks, S. Guadarrama et al., "Long-term recurrent convolutional networks for visual recognition and description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2015*, Boston, MA, USA, June 2015.
- [22] O. Ronneberger, P. Fischer, and T. Brox, "U-net: convolutional networks for biomedical image segmentation," *Lecture Notes in Computer Science*, vol. 9351, pp. 234–241, 2015.
- [23] J. Fu, K. Singhrao, X. S. Qi, Y. L. Yang, D. Ruan, and J. H. Lewis, "Three-dimensional multipath DenseNet for improving automatic segmentation of glioblastoma on pre-operative multimodal MR images," *Medical Physics*, vol. 48, no. 6, pp. 2859–2866, 2021.