TEST Engineering & Management

July - August 2020
ISSN: 0193-4120 Page No. 5699 - 5706

Image Edge Detection Using Parallel Depth Learning

Peng Xu[1,*] Xiaohui Zhang[1], Zhihua Zhai[2]

[1]School of Information Engineering, Yellow River Conservancy Technical Institute, KaiFeng, Henan, 475000, China

[2]Yellow River Conservancy Technical Institute, Kai Feng, Henan,475004,China

Article Info
Volume 83
Page Number: 5699 - 5706
Publication Issue:
July - August 2020

Article History
Article Received: *25 April 2020*
Revised: *29 May 2020*
Accepted: *20 June 2020*
Publication: *28 August 2020*

Abstract

With the development of scientific information technology, image processing and computer vision technology have gradually become an important research direction. At the same time, deep learning has been greatly developed and become a technological change. Various kinds of deep neural networks are constantly emerging. Therefore, the application of deep learning methods and probing neural networks to edge detection has become a new trend. In this paper, the traditional differential operator edge detection method, deep learning and deep neural network theory are systematically expounded. On this basis, the advantages and disadvantages of edge detection operator are analyzed, and the edge detection technology based on parallel depth learning is introduced. Through the analysis and validation of the experimental results, the parallel depth learning edge detection technology has a strong target understanding ability, is more suitable for extracting the contour of the target object, effectively suppresses the edge of non-target object, and is helpful for subsequent target detection and analysis.

Keywords: *Image edge detection; Parallel depth learning;*

1. Introduction

Image edge detection has always been one of the research hotspots in the field of image processing and computer vision. Edge information is one of the important features of objects, and it is also an important basis for target recognition. Image edge contains most of the information of the image. Accurate image edge detection results can not only greatly reduce the amount of information to be processed when analyzing the image, but also effectively obtain the boundary structure of the image. Edge is the part of the image where the local mutation occurs. It represents the end of the feature area of an old thing and the beginning of the feature area of another thing. For the edge itself, it contains a lot of intrinsic information, which can provide important core features for image recognition. Essentially, the edge is a discontinuous place, an observable external form of expression.

The basic steps of image edge detection are: (1) filtering; (2) enhancement; (3) detection; (4) localization; as shown in Figure 1. After years of rapid development, edge detection technology has formed different methods and theories, and has different usage scenarios for different application environments. In the future technological development, as a basic theory, edge detection will get more development.

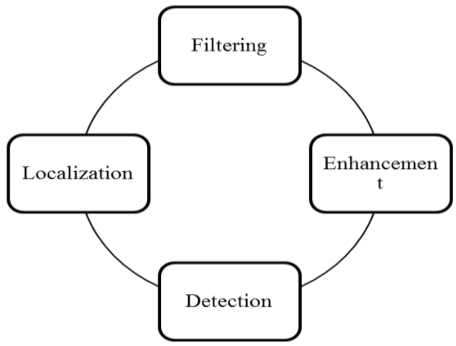

Figure 1. Basic steps of image edge detection

2. Application of deep learning in image edge detection

2.1 Deep learning and deep neural network

BP neural network is mainly composed of three layers, including input layer, single hidden layer and output layer. The structure of BP neural network is shown in Figure 2.

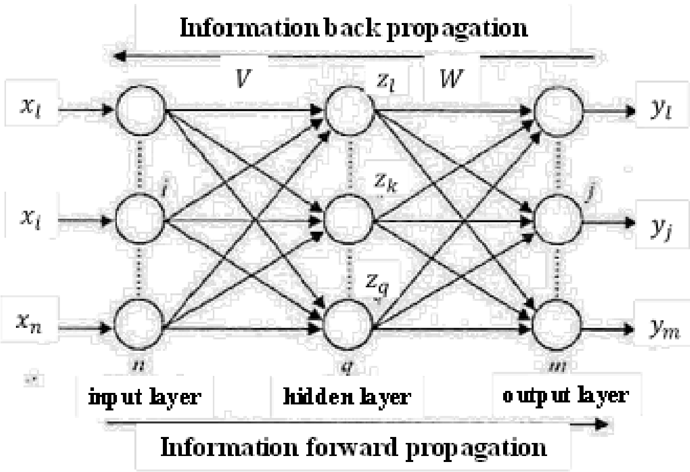

Figure 2. Structural of BP neural network

Input signal is calculated through hidden layer and pre-set weight value. After output, it is compared with input signal. If it is larger than the initial threshold, the weight value is modified until the output results meet expectations and the training process is completed. In the training process, the weight coefficients are constantly modified and improved, so that the BP network meets expectations, and the testing process can be started. The testing process is that the weighted coefficients of input data that need to be tested are gradually stabilized in BP network, and the desired output results can be obtained after detection.

The structure of deep learning is similar to that of BP neural network, except that there are many hidden layers in the middle layer, that is, there are many hidden layers between the input layer and the output layer. In this way, there will be more weight coefficients and thresholds between the hidden layers, between the input layer and the first hidden layer, between the output layer and the last hidden layer, and the amount of calculation will increase geometrically.

The most commonly used deep neural network models are CNN, DBN and SAE. Convolutional neural network is a general term, which is a set of multi-layer neural networks. It is especially suitable for plane data, such as images and videos. CNN is the first truly successful deep learning method. The hierarchical structure of multi-layer network is trained in a robust way. Convolutional neural network is improved in general feedforward and backward propagation training. Convolutional Neural Network (CNN) proposes a framework for deep learning, which requires minimum data pre-processing information to be disseminated through different layers of the network. In order to obtain significant features of the observed data, digital filtering is applied at each layer. This method provides invariance of translation, scaling and rotation. Local sensory visual field allows neurons or processing units to access some basic features, such as directional edges or directional angles.

Deep belief network (DBNs) is a probabilistic generation model based on generation model and discriminant model, which is compared with traditional neural network, and its feasibility and advantages are proposed. The generated model provides the joint probability distribution of observation data and labels, and promotes the estimation of P (observation label) and P (label observation), while the discriminant model is limited to the latter, that is, the estimation of P (label observation).

Generating model and discriminant model, both of which are essentially supervised learning methods, and supervised learning methods are divided into discriminant method and generating method. Discriminant method is to obtain a judgment formula or method under some known limited conditions, regardless of the process of data sample generation in the judgment process, only focusing on the results of judgment. The method of generating data is to obtain a general formula in a wider sense under limited conditions, which is a process of generating data samples through learning, and then to obtain a judgment method from the general formula under specific conditions. The latter part is just like a judgment method.

DBN is a typical neural network composed of Restricted Boltzmann Machines (RBM). The structure of DBN network is shown in Figure 3.

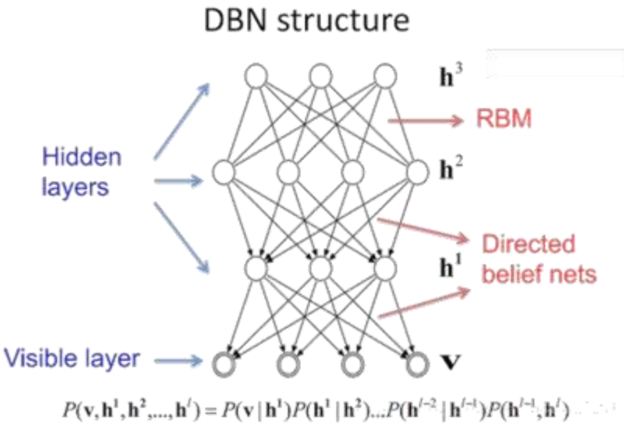

Figure 3. Structure of DBN network

Stack self-coding network and DBN are similar in structure, and they are also unsupervised deep learning networks. Unlike DBN, DBN uses RBM as its structural unit, while SAE uses auto-encoder (AE) as its structural unit. The structure of self-coding is shown in Figure 4.

In order to achieve a minimum reconstruction error between the original signal and the reconstructed signal, the original signal is first coded by the encoder and then decoded by the decoder. After this process, some unnecessary redundant information will be removed, and the most essential features will be preserved, thus reducing the data to be processed.

On the basis of the self-encoder, some pre-processing methods are added to improve it, mainly de-noising self-encoder, as shown in Fig. 5. In order to achieve a minimum reconstruction error between the original signal and the reconstructed signal, the original signal is first coded by the encoder and then decoded by the decoder. After this process, some unnecessary redundant information will be removed, and the most essential features will be preserved, thus reducing the data to be processed.

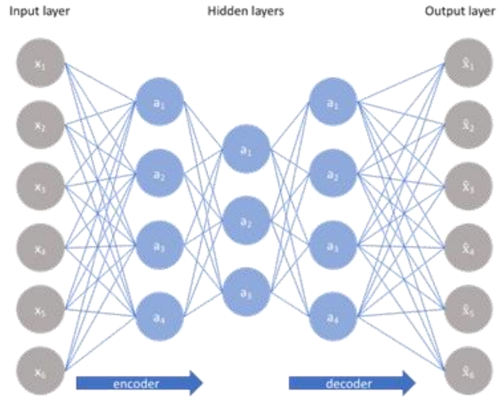

Figure 4. Schematic diagram of self-encoder structure

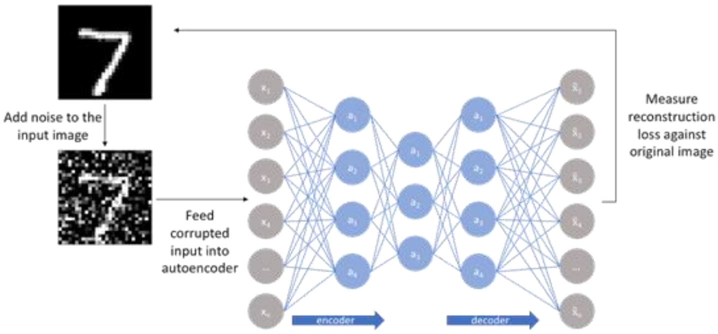

Figure 5. Noise reduction self-encoder architecture

On the basis of the self-encoder, the de-noising self-encoder adds the de-noising processing, and then the hidden layer feature representation is obtained by the encoder processing. Then the reconstructed signal is obtained by the decoder, and the reconstructed error is obtained by comparing the reconstructed signal with the original input. The aim is also to minimize the reconstruction error so that it can adapt to certain noise interference and improve the adaptive ability.

2.2 Application of deep learning in image edge detection

With the rapid development of deep learning technology, the application of deep neural network in edge detection has become a reality. With the development of deep learning technology, edge detection can be applied to more complex scenes.

The whole nested edge detection method is a new representative depth network algorithm for edge detection. This network structure is called Holistically-Nested Network (HED). HED is a

Published by: The Mattingley Publishing Co., Inc.

July - August 2020
ISSN: 0193-4120 Page No. 5699 - 5706

method of edge detection which makes full use of CNN and depth monitoring network. In the process of detection, the problem of blurring edge and target boundary can be solved by automatically learning rich hierarchical representations (guided by depth monitoring). The existing network structure of multi-scale and multi-level learning for other HED and HED is shown in Figure 6.

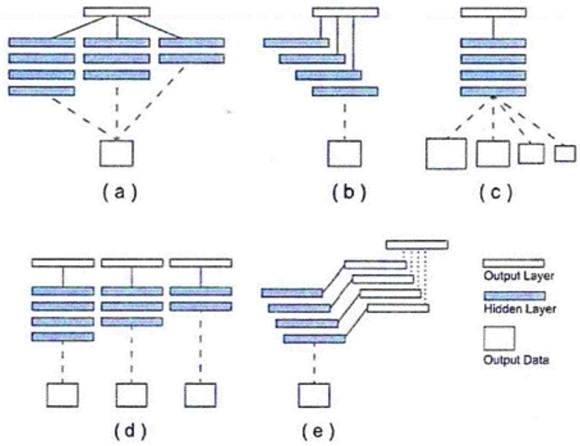

Figure 6. Multi-scale and multi-level network structure diagram

Among them, (a) multi-stream architecture, (b) skip-layer architecture. (c) It is a single model operation structure (a single model running on multi-scale input). (d) Separate training structures of different networks. (e) HED network structure, the main improvement is to add a side output layer behind each convolution layer, each side output layer output an edge map. Then, the output of each side output layer (i.e. edge map) is processed by a weighted fusion layer, and the required results are output by a pre-set calculation method.

HED method uses holistically to make the result of edge detection holistic, image-to-image and end-to-end. The purpose of using nested is to express clearly that in the process of edge detection, the output result is continuous learning and inheritance. HED algorithm is suitable for edge detection in complex situations. Different thresholds can be set to obtain different accuracy results at different time and space costs. When the picture is complex, HED has better detection effect.

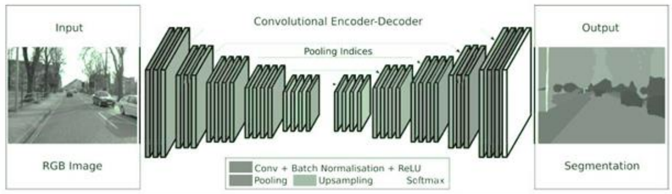

Figure 7. Full convolutional encoder-decoder network

Object Contour Detection with Fully Convolutional Encoder-Decoder Network is a high-level edge detection method, which is target contour detection. Full convolution encoder-decoder network edge detection method is different from the previous low-level edge detection algorithm, focusing on the detection of higher-level object contours. The full convolution Encoder-Decoder network is shown in Figure 7. This algorithm mainly uses a simple but very effective method to achieve the goal of target contour detection, that is, coding-decoding network. Encoder process: Make the image convolute with Maxpooling continuously, and get the intermediate result. Decoder process: Let the intermediate results be anti-pooling process, and continue deconvolution to get the image detection results. The results of image edge detection obtained by coding-decoding network are better than other methods, but the detection results are not clear compared with HED algorithm.

Relaxed Deep Supervision with Convolutional Neural Networks (RDS-CNN) is an edge detection method. RDS-CNN refers to an image with positive labels and relaxed labels, which is obtained by adding relaxed depth supervision to the CNN network and passing the input image through the CNN network each time. Finally, the image with positive labels and relaxed labels is used. The results of each convolution neural network are fused to get the output after fusion.

Compared with the whole nested edge detection method, the convolution neural network edge detection method under relaxation depth supervision has better detection effect on the small edges of complex images, but the HED algorithm has a

Published by: The Mattingley Publishing Co., Inc.

thicker, more obvious and better detection effect than the RDS-CNN algorithm.

By analyzing and comparing several recent deep learning networks applied to edge detection algorithms, such as HED algorithm, full convolution encoder-decoder network edge detection method, RDS-CNN method and so on. It can be concluded that the full convolution encoder-decoder network edge detection method has some disadvantages compared with the other two methods. RDS-CNN method has better detection effect for the small edges of complex scenes, but the HED detection results are more rough and obvious.

3. Research status of parallelization of deep neural networks

The application of deep neural network can be divided into training process and reasoning process. When the training process is designed concurrently, multi-core and multi-core technology (GPU, MIC) is used to accelerate the training process in parallel on a single node. For the training process of large-scale deep neural network which cannot be completed by a single node, the data communication between nodes is usually completed by combining MPI or Spark, and the whole model is trained by distributed system. In parallel acceleration of reasoning process, the current research focuses on acceleration through dedicated accelerator or FPGA, which has the advantages of low power consumption and fast speed.

3.1 GPU and MIC

GPU is used to accelerate the training process of deep neural network algorithm. Usually, CUDA or OpenCL is used to transplant the algorithm to GPU, and the algorithm is accelerated by data parallel or model parallel, or by combining the two methods.

When using MIC to speed up the training process of deep neural network in parallel, OpenMP parallel programming language is generally used. The loop part of the algorithm is multithreaded and SIMD parallel through OpenMP and vectorised instructions. There are two programming modes commonly used in MIC, one is uninstall mode: one is to run part of the program on the CPU, the other is to uninstall the program on the MIC and execute in parallel, in which the required data is also transmitted to MIC through PCI-E. The other is a native mode in which MIC runs all programs independently.

3.2 ASIC and FPGA

When designing a special accelerator, we usually design the corresponding hardware circuit for it by analyzing the characteristics of the algorithm, make full use of the parallel parts of the algorithm, and assign specific computing elements for the corresponding operation, so as to achieve higher acceleration effect. For example, the output of each neuron in the convolution layer and the connective layer is a multiplication and accumulation operation by several weights and corresponding input values. A certain size array of processing units can be designed. Each processing unit is responsible for the operation of multiplication. By transforming the input feature graph and weight matrix into data stream, each processing unit can achieve high parallelism by completing its multiplication operation. Therefore, the special accelerator has a high acceleration ratio compared with CPU and GPU.

As a common reconfigurable device, FPGA is programmed by programmable interconnected configurable logic block (CLB) matrix. The typical structure of the FPGA is shown in the figure. It consists of three components: I/O Block for defining input and output signals, Interconnect Resource for customizing the connection relationship between logical blocks, and Logic Block for realizing the logic function of circuits. The architecture of the FPGA is shown in Figure 8.

Published by: The Mattingley Publishing Co., Inc.

July - August 2020
ISSN: 0193-4120 Page No. 5699 - 5706

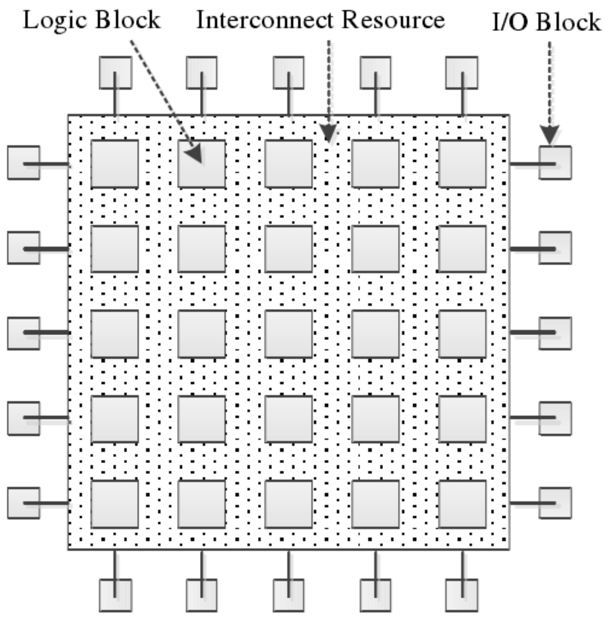

Figure 8. Architecture of FPGA

The number of network layers and the complexity of the model of deep neural network algorithm are increasing year by year. It is the key to design efficient algorithms. It is the trend to use low-precision weights and sparse network connections. The binary neural network algorithm, which only uses 1 and -1 as weights, has developed rapidly due to the requirement of reducing storage space and memory bandwidth.

3.3 Multi-node parallelization

The early deep neural network multi-node parallelization was mainly computed on CPU cluster. Although there are many hardware architectures and products for in-depth learning, because of the good software ecosystem of CUDA, open source in-depth learning software currently supports cuDNN, which makes most of the in-depth learning applications run on NVIDIA GPU. With the application of in-depth learning in more and more fields, in addition to GPU, considering the power consumption, volume, price and development cycle of devices running in-depth learning, the research of in-depth learning based on FPGA and ASIC has gradually become a new hotspot.

4. Experimental test and analysis

4.1 Experimental settings

The experimental software environment is 64-bit WIN10 operating system. Microsoft Visual studio 2015 is used as compiler tool, simulation tool is MATLAB 2015b software, and in-depth learning tool is NVIDIA GPU toolkit CUDA 8.0 and cuDNN. CUDA and cuDNN are parallel computing tools specially developed by NVIDIA for working on GPU for deep learning networks. The CAFFE package of the deep learning network tool library is selected, which includes the relevant library functions needed to establish the deep learning network. The hardware environment of the experiment is Intel CPU i7 2.8GHz, NVIDIA GPU GeForce GT 720M2G and 8GB memory.

SET14 is selected as the test set, which covers several categories of people, animals and natural scenes, and can effectively verify the performance of edge detection algorithm. According to the classical design method of filter size in deep convolution networks, the first layer is a 9*9 size filter, that is, $f_1 = 9$; the second layer is a 1╳1 size filter, that is, $f_2 = 1$; And the third layer is a 5╳5 size filter, that is, $f_3 = 5$. Noise is added to the training input image to improve the robustness of the network training.

4.2 Experimental results and analysis

The Zebra edge detection effect is shown in Figure 9. Comparing Figure 9(b) with Figure 9(c), it can be found that Figure 9(b) mainly detects the contour edge of the image by using the algorithm proposed in this paper, which can basically reflect the contour characteristics of the image content, while there is no more scene edge information. Figure 9(c) uses Canny algorithm for edge detection, which has two main shortcomings: on the one hand, the image has too many detail edges; on the other hand, both the target and the background are detected, which makes it more difficult to extract the target. Through comparison, it is found that this algorithm can only deal with the information concentration area, thus improving the detection efficiency.

Published by: The Mattingley Publishing Co., Inc.

July - August 2020
ISSN: 0193-4120 Page No. 5699 - 5706

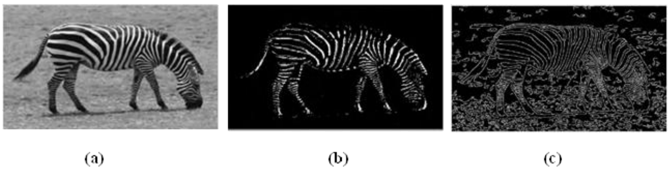

Figure 9. Zebra edge detection effect

Coast edge detection effect is shown in Figure 10. Comparing Figure 10 (b) with Figure 10 (c), it can be found that Figure 10 (b) mainly detects the contour edge of the ship by using the algorithm, which can basically reflect the contour characteristics of the ship without much scene edge information. Figure 10 (c) Canny algorithm is used for edge detection. Because there is more edge information on the water surface, the effect of Canny algorithm is worse. It is difficult to separate the edge of the boat from the water surface, which will affect the accuracy of subsequent target extraction and recognition. The algorithm shown in Figure 10 (b) can effectively improve the accuracy of target extraction because it can separate the boat from the water surface.

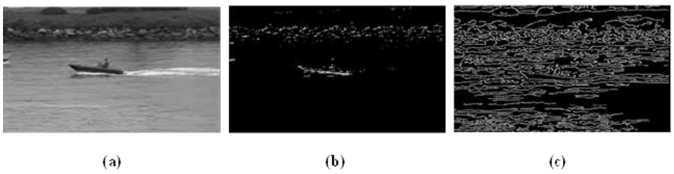

Figure 10. Coast edge detection effect

The experimental results above show that the parallel depth learning algorithm for image edge detection firstly establishes a three-layer convolution network, and through deep learning of large-scale image data sets, the convolution network learns the edge structure features of natural images and obtains relevant training parameters. Then, the training parameters are used to guide the model to detect the edge of the test image, so that the target contour and scene edge can be distinguished better. Compared with the traditional edge detection algorithm, the image edge detection algorithm based on parallel depth learning has better target understanding ability in mixed images of target and scene. It is more suitable for extracting the contour of target object, effectively suppressing the edge of non-target object, and is helpful for subsequent target detection and analysis.

5. Challenges and prospects of deep learning parallelization

Deep neural network has brought a new wave of machine learning, which has been widely valued from academia to industry. However, due to the complexity of the deep neural network algorithm, the number of iterations, and the high computational complexity, there are some challenges and bottlenecks in the parallelization of deep neural network. The future development of deep neural network parallelization has the following directions.

5.1 Automatic partition of tasks in deep neural network model parallel

The deep neural network model parallelization mainly aims at the designed neural network structure, which is divided by hand and mapped to different computing devices. Because of the inaccurate estimation of the running time of the task load, the manual network partitioning can easily lead to the unbalanced load on the computing nodes. In order to achieve automatic partitioning of network models and load balancing, it is also faced with the difficult problem of how to construct a precise performance model of deep neural network operators and design task scheduling algorithm.

At present, deep neural networks are mainly represented by computational graphs. In order to map computational tasks in computational graphs to multiple parallel computing hardware systems, we first need to construct a performance model for deep neural network operators in computational graphs. Considering the impact of hardware architecture on the running time of programs, we analyze the execution process of neural network operators and obtain the factors affecting their program performance. Finally, the performance model of each operator in the deep neural network is constructed. By solving the performance model, the execution time of each operator on the parallel computing device is obtained. Then, according to the structure characteristics of the deep neural

Published by: The Mattingley Publishing Co., Inc.

network, a reasonable and effective task scheduling algorithm is designed. The model partitioning strategy of the deep neural network is obtained through the scheduling algorithm. According to the model partitioning strategy, the computational tasks in the computational graph are mapped to the hardware system of parallel computing.

5.2 Parallel deep learning neural networks trends

For the future development trend of data parallelism in deep neural networks, we can start from two directions: first, design a distributed random gradient descent algorithm with fast convergence speed and low communication cost; second, solve the communication bottleneck problem between different nodes in the cluster.

6. Conclusions

The basic theory of traditional differential operator is introduced. On this basis, the advantages and disadvantages of traditional differential operator are summarized, and the edge detection technology based on parallel deep learning is introduced. Secondly, the theory of deep learning edge detection method is discussed systematically and deeply. On the basis of introducing the origin and development of deep learning and neural network, as well as the structure composition and classification of deep learning neural network, the working principle and network structure of some deep learning neural networks are mainly introduced, including CNN, SAE and DBN, and the application of deep learning in edge detection technology is analyzed. Including the whole nested edge detection algorithm, the full convolution encoder-decoder network edge detection method, and the convolution neural network edge detection method under the relaxation depth supervision. Based on the analysis and comparison, the parallel depth learning algorithm is emphatically studied. The experimental results show that the method has a strong target understanding ability, is more suitable for extracting the contour of the target object, effectively suppresses the edges of non-target objects, and is helpful for subsequent target detection and analysis.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016. 3.

[2] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015. 2.

[3] M. Maire, S. X. Yu, and P. Perona. Reconstructive sparse code transfer for contour detection and semantic labeling. In ACCV, 2014.

[4] P. Pinheiro and R. Collobert. Recurrent convolutional neural networks for scene labeling. In ICML, 2014.

[5] S. Cohen, H. Lee, and M. H. Yang. Object contour detection with a fully convolutional encoder-decoder

[6] network. In CVPR, 2016.

[7] R. Vemulapalli, O. Tuzel, M. Y. Liu, and R. Chellapa. Gaussian conditional random field network for semantic segmentation. In CVPR, 2016.

[8] Q. Shan, B. Curless, Y. Furukawa, C. Hernandez, and S. Seitz. Occluding contours for multi-view stereo. In CVPR, 2014.

[9] J. Yang, B. Price, S. Cohen, H. Lee, and M. H. Yang. Object contour detection with a fully convolutional encoder-decoder network. In CVPR, 2016.

[10] S. Xie and Z. Tu. Holistically-nested edge detection. In ICCV, 2015.

[11] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[12] M. Y. Liu, S. Lin, S. Ramalingam, and O. Tuzel. Layered interpretation of street view images. In RSS, 2015.

[13] G. Bertasius, J. Shi, and L. Torresani. Semantic segmentation with boundary neural fields. In CVPR, 2016