Edge detection using neural network committee

Diego Oliva[1], Demin Anton[2], Demeshko Maria[2]
diego.oliva@cucei.udg.mx | ad@tpu.ru | mvd7@tpu.ru
[1]Depto. de Ciencias Computacionales, Universidad de Guadalajara, CUCEI, Av. Revolucion 1500, Guadalajara, Jal, Mexico
[2]National Research Tomsk Polytechnic University, pr. Lenina 30, Tomsk, Russia

This article considers edge detection using neural network approach, and describes the integration of neural networks into a committee. The novelty of the proposed technique comes from two ideas: firstly, since the training set describes a solution, the proposed algorithm is able to detect multidirectional edges on noised image. Secondly, to increase the accuracy of detection, neural networks were combined into a committee.

Keywords: *edge detection, neural network, neural network committee, committee classification, digital image.*

1. INTRODUCTION

The necessity to detect edges is one of the crucial task in computer vision and image processing systems. Commonly it is used for applications such as segmentation or object recognition [1]. Edges are features that define the area of an object in digital images. Basically, an edge is a boundary between two homogeneous regions [2]. The gray level properties of the two regions on either side of an edge are distinct and exhibit some local uniformity or homogeneity among themselves.

There are different approaches used for edge detection and it is important to emphasize that each one of them can be useful in specific cases defined by the scene. This situation exists since edges can be defined in different ways for different situations. One of the definitions, which have many applications, is the local variation of the brightness in the image [3].

On the other hand, several classic methods of edge detection use the gradient evaluation. One of the first methods was proposed by L. Roberts [5], it is based on cross matrix operator, which contains differences between neighbor elements. In this sense, J. Prewitt introduced an operator based on central difference [5], that is well-known and extensively used for edge detection. Another interesting technique to detect edges in digital images is the discrete Laplace operator, its kernel is evaluated with discrete partial derivatives [6].

Some other approaches consider filters with weighted kernels used for decreasing the noise [5]. An example of this kind of technique was developed by J. Canny [4]. It is an edge detection method which includes gradient evaluation but in addition it uses a preprocessing step (blurring) and non-maximum suppression. Morphological operations can also be applied for the described problem [7]. This approach employs the difference that exists between the original image and its erosion output. The main disadvantage of this method is that it can be applied only for binary images. Meanwhile, several studies demonstrate that the usage of wavelet transform for edge detection is an interesting tool and provide competitive results [8, 9].

All the methods previously described represent interesting alternatives for edge detection. However, the main problems of all of them are the computational effort required and the lack of accuracy presented on the detection [10]. This article considers edge detection using neural network approach, and describes the integration of neural networks into a committee. The novelty of the proposed technique comes from two ideas: firstly, since the training set describes a solution, the proposed algorithm is able to detect multidirectional edges on noised image. Secondly, to increase the accuracy of detection, neural networks were combined into a committee.

The remainder paper is organized as follows: Section 2 presents the architecture of the Neural Network. In Section 3 the training process is explained. Section 4 explains the bagging procedure used for networks committee. In Section 5 the process to form the inputs of the Neural Network is presented. The experimental results are presented in Section 6. Meanwhile, Section 7 shows a comparative study of the proposed approach. Finally, in Section 8 some conclusions are discussed.

2. NEURAL NETWORK STRUCTURE

In this article multilayer feedforward neural networks are used. In this kind of networks the information moves only in one direction from the input nodes through the hidden nodes and to the output nodes. Signal transmission occurs as follows: the sum of the products of the weights and the inputs is calculated in each node. Values calculated in output neurons become network outputs and can represent significant information about input vector.

In order to choose optimal network configuration several test networks were trained. These networks had different configurations:

1. 25 input neurons, 10 hidden neurons, 1 output neuron;
2. 25 input neurons, 25 hidden neurons, 1 output neuron;
3. 25 input neurons, 30 hidden neurons, 1 output neuron;
4. 25 input neurons, 50 hidden neurons, 1 output neuron.

Number of inputs is 25 for all of the experiments because of chosen neighborhood. Every input vector represents the set of values corresponding to pixels from neighborhood with size 5×5. Way of formation of these values is described in Section 5.

Experiments with these networks included training with the study images described in Section 3 and detecting edges for one image. Results were analyzed visually. Network with 30 hidden neurons detected edges better than others.

Selected structure is shown in the Fig. 1.

The publication of the conference proceedings was supported by the RFBR, grant No. 18-07-20045\18

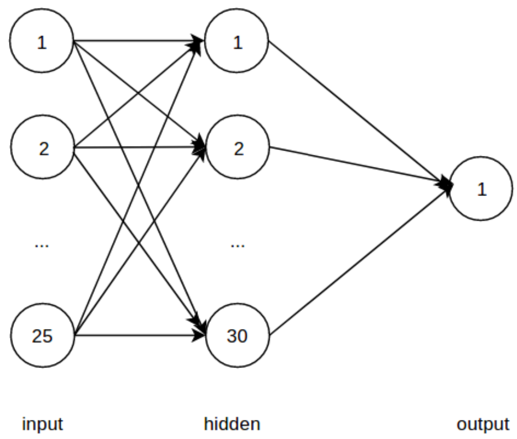


Figure 1. Neural network structure.

In accordance with experiment neural network includes 25 input neurons, 30 hidden neurons and 1 output neuron. The value of output neuron will be set to zero if current pixel belongs to background or object, and will be set to one if current pixel belongs to the edge.

Described detection system includes three networks with such structure.

3. NETWORK TRAINING

In order to properly train the neural network, a dataset is generated that contains different samples of edges. The images on the training set were prepared in such a way that they have edges of different "steepness". Also, they are contaminated with noise to achieve appropriate noise resistance. All these operations were performed with the help of GIMP editor.

The samples included in the training dataset are presented in Fig. 2–4.

Figure 2. Couple of study images: original and marked (horizontal edges).

Figure 3. Couple of study images: original and marked (vertical edges).

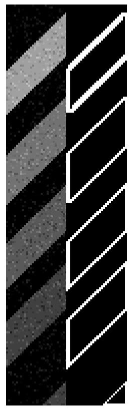


Figure 4. Couple of study images: original and marked (diagonal edges).

In Fig. 2–4 the marked images were prepared with the help of the Canny detector and after that were modified manually to achieve necessary accuracy. In this way, each image is a multiple testing set: the sets formed from the images in Fig. 2–3 both contains 1500 sub-images, the set formed from the images in Fig. 4 contains 6000 sub-images. Moreover, some of these sets contain edges and some do not: training with these images takes into account positive and negative examples.

4. BAGGING

Bagging is one of the committee classification methods. These kinds of methods are used in cases when it is necessary to define to which class the object belongs. In terms of edge detection committee classification approaches can determine whether current pixel belongs to the edge or to the object and background. The application of committee techniques theoretically is not worse than methods which use one classifier [11]. This rule is often observed in practice, but there are cases when the committee classification works worse than one classifier. Therefore, the usefulness of the committee classification for the solution of a particular problem is determined experimentally.

Bagging is one of the fundamental committee classification algorithms, in which the decision is made on the basis of averaging decisions of separate classifiers. Described solution uses three classifiers trained on various images (explained in the previous section). Their results are summarized and multiplied by empirical coefficient. If result is less than fixed threshold it means that current pixel belongs to background or object. Otherwise, if result is greater than threshold, current pixel belongs to the edge.

5. FORMATION OF INPUTS

In order to properly identify the optimal way of forming input vector, several experiments were conducted. Each experiment consists of using particular way of forming inputs and submitting this data to neural network for training with the back-propagation algorithm. The resulting network was used for edge detection on different images and results of the detection were visually analyzed. The neighborhood had the same size (5×5) for all experiments. In first experiment input vector corresponded to the normalized values from the neighborhood of the point. This experiment gave torn thin edges which lead to three other experiments. Assuming that using differences will give better result, those experiments were conducted with usage of three types of differences: difference between central element and current element, difference between current element and its row neighbor, difference between current element and its column neighbor. Such differences are explained in Fig. 5–7.

Работа опубликована при финансовой поддержке РФФИ, грант 18-07-20045\18

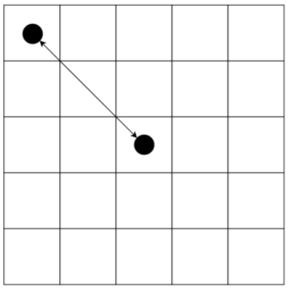


Figure 5. Difference between central and current element.

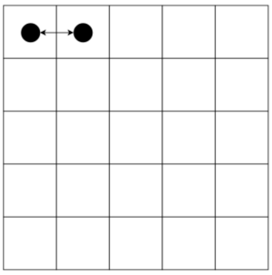


Figure 6. Difference between current element and its row neighbor.

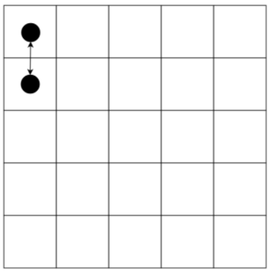


Figure 7. Difference between current element and its column neighbor.

It is interesting that using difference with column neighbor gave no edges. Using difference with central element gave best results (this method is in a sense similar to filter weighing), so it was the way of formatting inputs that was chosen.

6. RESULTS

This section presents some interesting results selected from the experiments. Results of edges extraction using the proposed method after the training and testing steps are presented in Table 1.

First row in Table 1 shows edge detection on the image of Rubik's Cube. Neural network detected most of the squares on the sides of the cube but did not detect blue ones. Presumably it happened because of less abrupt brightness transition.

Second row in Table 1 shows edge detection on the photo of pawn. It can be noticed that right corner of the pawn is not detected because of lightning, but overall outline of the chess figure is obvious.

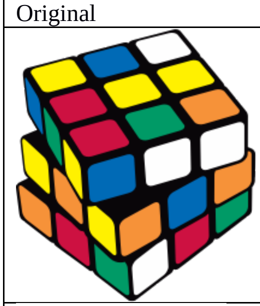
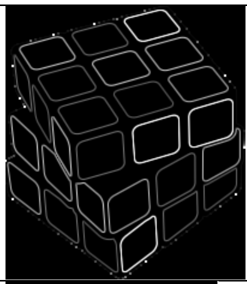
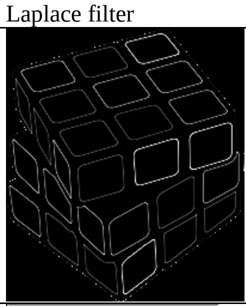
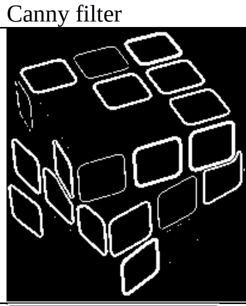
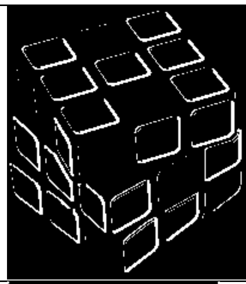
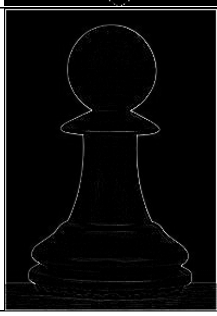
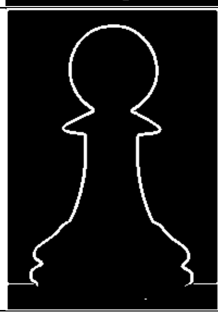
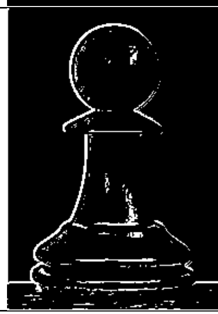
Last row in Table 1 shows result of edge detection on the image of sea wave. It is evident that detector identified many details and found direction of the waves.

It is obvious that the classifier detects edges of the objects. Sometimes it reacts to noise more than necessary (image with the pawn), but in other cases it can be useful (image with the wave).

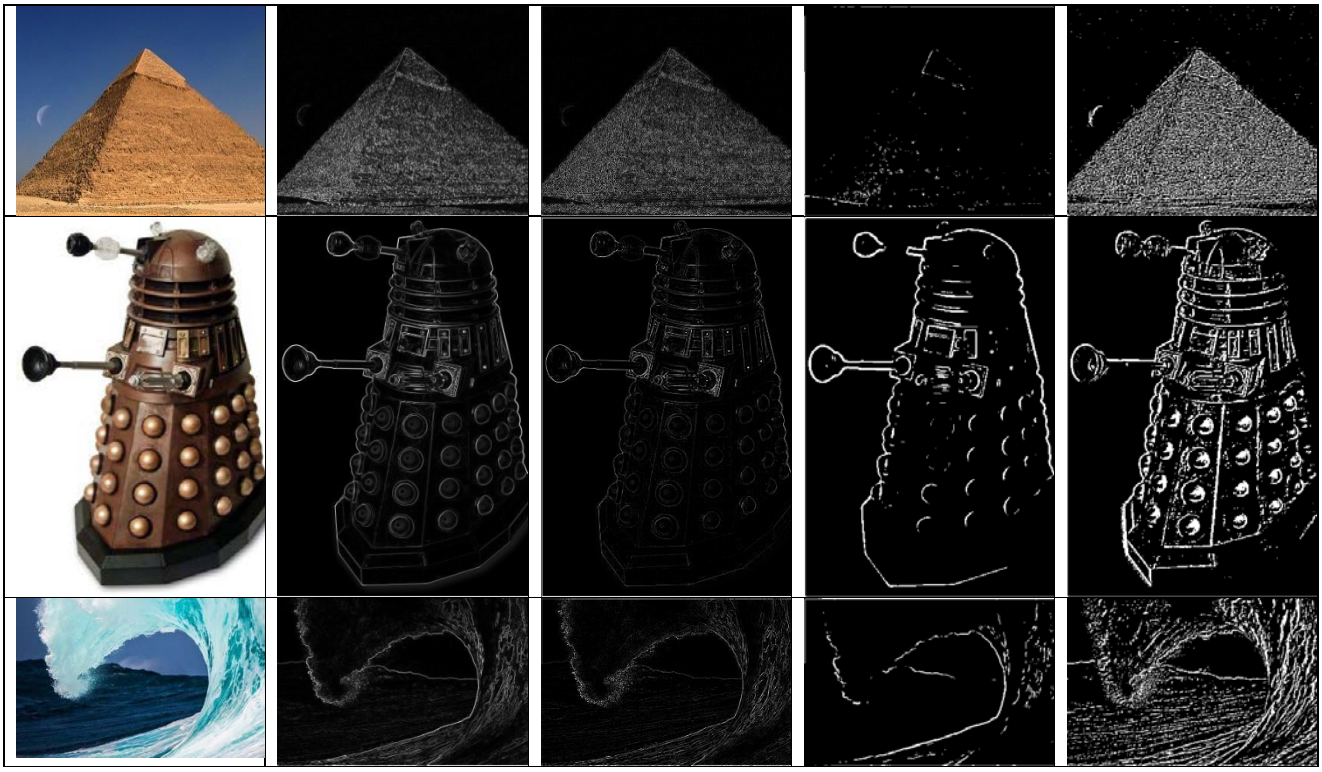
7. COMPARISON WITH OTHER METHODS

For a comparative study the Table 1 shows the results of edge detection considering different methods (explained in Section 1): the Roberts filter, the Laplace filter and the Canny filter.

Table 1. Comparison with other edge detection methods

Original	Roberts filter	Laplace filter	Canny filter	NN detector
				
				

The publication of the conference proceedings was supported by the RFBR, grant No. 18-07-20045\18



From Table 1 it can be noted that developed classifier is comparable with Canny detector in terms of thickness and accuracy. Also the last example shows that developed detector is sensitive to directed lines. This feature can be useful in detecting dynamic changes in image series. In this context, the image with Rubik's cube shows that sensitivity of developed detector is higher than Canny's but lower than Roberts' and Laplace's. On the other hand intensity of edges detected with described committee is higher than intensity of edges given by Roberts and Laplace filters.

8. CONCLUSION

Proposed classifier can be used as a preprocessing step in image segmentation and objects recognition algorithms. It detects edges which are as wide as edges detected by Canny filter. In cases where it is necessary to detect thin boundaries skeletonization methods can be applied. The application of the neural network committee allows usage of weak classifiers. Moreover, it is possible to replace sub-classifier for cases when it is necessary to detect edges with different characteristics (sub-classifier will be learned on different samples). Also edge detection can be applied to scientific visualization as a step of visualization quality assessment. Opportunity to replace sub-classifiers allows forming different requirements to the result. For example, in case of visualization of air flows the requirement for edge clarity between flows with different temperature can be brought. Committee composed of classifiers trained to detect edges of such clarity can find them and give the results to sub-system which makes the decision about the correspondence of the expected and actual result.

On the one hand, disadvantage of this classifier is its noise sensitivity, which is higher than Canny detector's sensitivity. But on the other hand, it can be useful in solving certain problems. For example, third image in Table 1 lost all features after Canny filtration, but described classifier detected all of the local changes in pyramid texture. It means that it is necessary to select edge detection tool in accordance with the task being solved.

The work was carried out within the TPU Competitiveness Enhancement Program.

9. REFERENCES

1. Belim S.V., Kutlunin P.E. Boundary extraction from images using clustering algorithm // Computer Optics. 2015. 39 (1). P. 119–124.
2. R.C. Gonzalez, R.E. Woods, Digital Image Processing, Pearson, Prentice-Hall, New Jersey, 1992.
3. Demin A.Yu., Dorofeev V.A. Parallelization research of Algorithm for detecting borders on the basis of graph representation // 12th International conference on actual problems of electronic instrument engineering (APEIE), 2014. // Novosibirsk, October 2-4 , 2014. Vol. 6. P. 232–236.
4. John Canny. A computational approach to edge detection // Pattern Analysis and Machine Intelligence, IEEE Transactions on, PAMI-8(6). P. 679–698.
5. Fisenko V.T. Computer processing and image recognition // ITMO University, 2008. P. 192.
6. Bui T.T.T., Spitsyn V.G. Analysis of methods of digital images edge detection // Reports of Tomsk State University of Control Systems and Radioelectronics, (2-2 (22)). P. 221–223.
7. Shapiro L., Sockman G. Computer vision // BKL Publishers, 2006. P. 752.
8. Antoshuk S.G. The edge selection of object images by double hyperbolic wavelet transform method // Electrical Machine-building and Electrical Equipment (65), 2005. P. 65–69.
9. Zhiznjakov A.L. Detection and analysis of shapes attributes of halftone images using multiresoluation representation // Methods and devices of information transmission and processing (7), 2006. P. 143–148
10. O.P. Verma, R. Sharma, M. Kumar, N. Agrawal, An Optimal Edge Detection Using Gravitational Search Algorithm, Lect. Notes Softw. 1, 2013. P. 148–152.
11. Bishop C. Pattern recognition and machine learning // Springer, 2006.

Работа опубликована при финансовой поддержке РФФИ, грант 18-07-20045\18