8-2019

Analysis of the CNN Algorithm in Target Recognition by Using the MSTAR Database

Ligang Zou
The University of Texas Rio Grande Valley

ANALYSIS OF THE CNN ALGORITHM IN TARGET RECOGNITION BY USING THE MSTAR DATABASE

A Thesis

by

LIGANG ZOU

Submitted to the Graduate College of
The University of Texas Rio Grande Valley
In partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2019

Major Subject: Mathematics

ANALYSIS OF THE CNN ALGORITHM IN TARGET RECOGNITION BY USING THE MSTAR DATABASE

A Thesis
by
LIGANG ZOU

COMMITTEE MEMBERS

Dr. Zhijun Qiao
Chair of Committee

Dr. Andras Balogh
Committee Member

Dr. Dambaru Bhatta
Committee Member

Dr. Jasang Yoon
Committee Member

August 2019

ABSTRACT

Zou, Ligang, Analysis of the CNN Algorithm in Target Recognition by Using the MSTAR Database. Master of Science (MS), August, 2019, 22 pp., 2 tables, 7 figures, 11 references, 11 titles.

With the rapid development of artificial intelligence technology and the emergence of a large number of innovative theories, the concept of deep learning is widely used in object detection, speech recognition, language translation and other fields. One of the important practices is target recognition in SAR images. Although it shows certain effectiveness in some researches, when using deep learning algorithm, there are still many problems that have not yet been solved. For example, people do not have a good understanding of how convolution works and the impact of convolution on the algorithm, although convolution works well in the CNN algorithm.

This thesis aims at analyzing the influence of the convolution in CNN algorithm. The goal can be achieved by controlling the convolution kernels. By controlling the amount of convolution kernels and the corresponding padding, the influence of convolution kernels will be determined. Then, the correctness of the above theories will be explained by conducting experiments using the MSTAR database.

DEDICATION

Without the support and encouragement of my family, my master's degree will not be completed. My mother and father were motivated wholeheartedly, and even though I was frustrated because of the terrible results of the experiment, they still encouraged and supported me to complete this degree. Thank you for your love and patience.

ACKNOWLEDGMENTS

Firstly, I would like to express my greatly sincere gratitude to my advisor Dr. Zhijun Qiao, for his guidance, tremendous patience and immense knowledge throughout my graduate studies. He provided me continuous support of my master's study, related research and great opportunity to take part joint research. His guidance helped me a lot in research and writing of this thesis. The most valuable lesson that I learned from him was on how to become an independent learner. Besides my advisor, my sincere thanks also go to my thesis committee members Dr. Andras Balogh, Dr. Dambaru Bhatta and Dr. Jasang Yoon for their insightful comments and encouragement which helped to ensure the quality of my intellectual work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Background of the Convolution Neural Network (CNN)

Neural Network (NN) technology was mentioned for the first time in the 1950s- 1960s. Called perceptron, NN has an input layer, an implicit layer, and an output layer. Usually, eigenvectors are input to the input layer and transformed through the hidden layer, then some classification results will be obtained and be presented at the output layer. A single-layer perceptron proposed by Frank Rosenblatt has a serious problem, that is, it can't do anything with a slightly more complex function (such as the most typical one, XOR operation). This shortcoming was not solved by the multi-layer perceptron invented by Rumelhart, Williams, Hinton, LeCun, etc. until the 1980s[4]. The multi-layer perceptron solves the defect that can't simulate the XOR logic before. It also makes the network more capable of portraying complex situations in the real world.

Multilayer perceptrons can get rid of the constraints of early discrete transfer functions, use continuous functions such as sigmoid or tanh to simulate the response of neurons to excitation. Training algorithm, now called neural network, can use the backpropagation (BP) algorithm invented by Werbos. BP algorithm is also called BP neural network. However, BP neural network (multilayer perceptron) has some problems such as non-convex optimization (local optimization), Gradient Vanish, over-fitting, etc. As the number of neural network layer increases, local optimization and Gradient Vanish will become more and more serious, even worse than shallow neural networks.

In 2006, Hinton used the pre-training method to solve the problem of local optimal solution, pushing the hidden layer to the 7th layer, which makes the neural network has a ”depth” in the true sense, leading to uncovering the upsurge of deep learning, and the DBN, CNN, RNN, LSTM, etc. At the same time, in order to overcome the Gradient Vanish problem, transfer function such as ReLU replaced sigmoid, resulting to the basic form of CNN[9]. However, from a structural point of view, there is no difference in a fully linked multi-layer perceptron.

1.2 Synthetic Aperture Radar (SAR)

Synthetic Aperture Radar (SAR) is a microwave imaging system that can obtain high-resolution radar images 24/7 under any weather condition. Emerging from 1950s, SAR is one of the most popular remote sensing system that have been widely used around the world. Amongst may application, target recognition using SAR images is one of the most important one and has been extensively researched nowadays.

1.3 Overview

This paper will analyze the experiment based on the characteristics of CNN algorithm, so that it can get better results in SAR image processing. The rest of the paper is organized as follows:

1. In the Chapter 2, a brief description of the CNN algorithm has been analyzed, and the reason for using the CNN algorithm (especially the convolution kernels) is explained.

2. In the Chapter 3, the MSTAR Data Base is tested using the CNN algorithm, and the experimental data and results are analyzed to prove the correctness of the conclusion.

3. In the chapter 4, the effect of padding on the convolution layer will be analyzed, and the experiment will be performed by controlling padding to verify the correctness of the analysis results.

4. The conclusion is drawn in the last chapter.

CHAPTER II

CNN ALGORITHM AND ITS CONVOLUTION KERNELS

In this chapter, the CNN algorithm will be introduced, and the most special part, the convolution layer will be analyzed, and the experiment will be based on the analysis result, trying to explain whether the analysis result is correct.

2.1 Structure of the Convolution Neural Network

The CNN algorithm can be regarded as one of the most successful special cases of Deep Neural Network (DNN). It has a wide range of applications in the field of image recognition for deep learning. The CNN algorithm includes an input layer, an implicit layer, and an output layer. The hidden layer includes a convolution layer, and the pooling layer is fully connected. The general flow is shown in the following Figure 2.1.

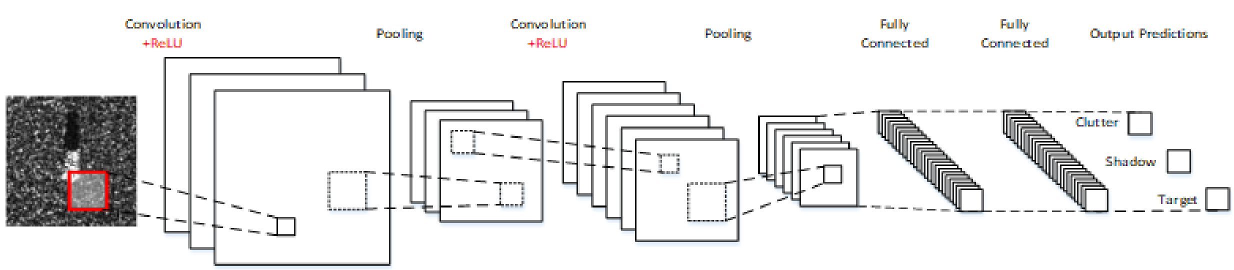


Figure 2.1: Flow Chart of the Convolution Neural Networks.

In the above process, the Fully Connected Layer (FC) located behind the convolution layer and the pooling layer is a commonly known DNN structure. The convolution layer and the pooled layer are unique to CNN, and the convolution layer has an activation function corresponding to it, that is, the ReLU function mentioned above, which can be summarized into the following format by the following equations

$$ReLU(x) = max(0, x) \tag{2.1}$$

where x is the input to the neuron.

Compared with Sigmoid functions, ReLU has the advantages of saving computation and preventing Gradient Vanish. At the same time, because the ReLU function will cause the neuron output to be less than 0, resulting the reduction of interdependence of parameters and solves the problem of over fitting. The pooling layer does not have an activation function[6][7][2].

According to the concept of ”Feature Visualization” proposed by Matthew D Zeiler, Rob Fergus in 2014[5], the visualization of neural network features is gradually accepted by everyone, and the features are extracted through the convolution layer. The convolution layer enables the features of the image to be extracted. This paper analyzes the causes from a mathematical perspective.

In the calculus in mathematical, the convolution expression is defined as

$$S(t) = \int x(t-a)w(a)da \tag{2.2}$$

where x and w are two integrable functions, a is the variable of the integral, t is displacement of the function.

Its discrete form is:

$$S(t) = \sum_k x(t-k)w(k) \tag{2.3}$$

And when represented by matrix, the above expression can be expressed as

$$S(t) = (X * W)(t) \tag{2.4}$$

where X and W are matrices, $*$ is expresses the convolution.

If a two-dimensional convolution is used, the expression is

$$S(i,j) = (X * W)(i,j) = \sum_m \sum_n x(i-m, j-n)w(m,n) \tag{2.5}$$

In CNN, the convolution is slightly different from the convolution in mathematics, but it is similar, and for the two-dimensional convolution expression

$$S(i,j) = (X * W)(i,j) = \sum_m \sum_n x(i+m, j+n)w(m,n) \tag{2.6}$$

If convolute with the image, according to the above convolution formula, the matrix of the different parts of the input image and the elements of the corresponding position of the convolution kernel matrix are multiplied and then added[11][10][8]. The actual effect is shown as in Figure 2.2.

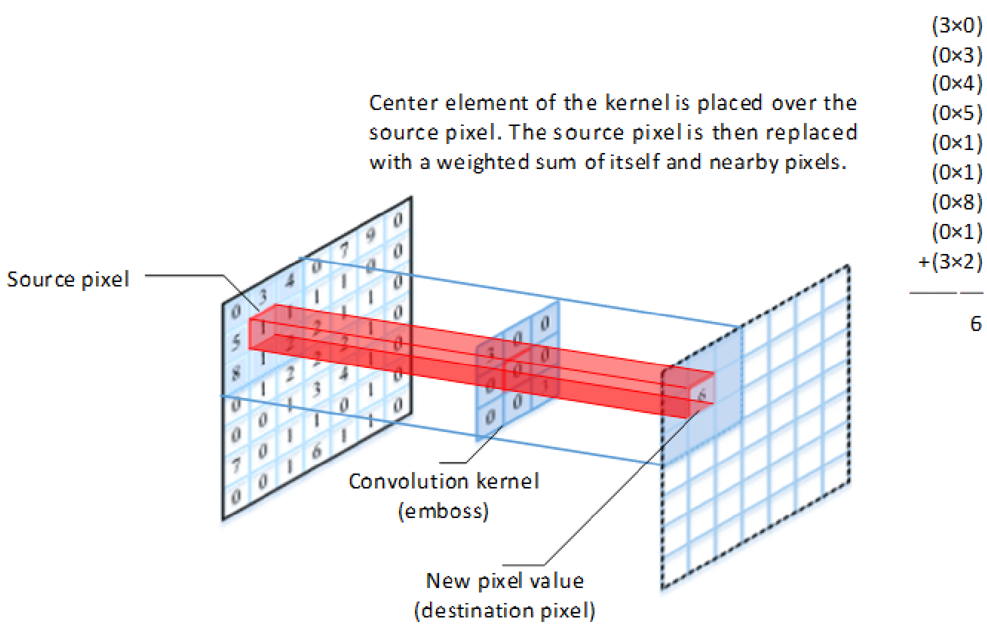
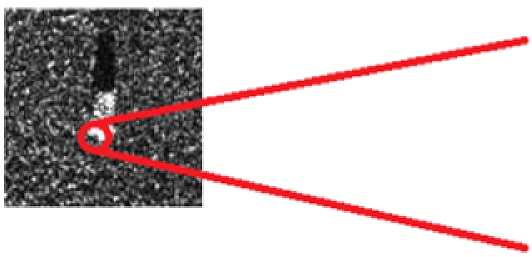


Figure 2.2: Convolution Operation in CNN.

It is known that a convolution layer is a layer that extracts features, that is, regardless of the number of layers of the convolution layer and the dimension of the convolution kernel, after convolution, the features of the image could be successfully extracted. The edge of the image is a more important feature of the image. Therefore, this paper analyzes the extraction of boundary features.

Converting the image into a matrix, it can be intuitively found that the value of the element of the matrix in the part of the boundary is significantly different from the value of the surrounding element (not necessarily the size of the element itself, as long as the difference is large) The difference between the matrix elements of the non-edge part is much smaller than the difference of the elements at the boundary.



73	44	95	55	22	54	80	68	63	54
82	141	137	74	44	43	48	40	33	60
144	183	110	53	50	30	25	28	5	41
66	58	94	41	54	19	22	51	11	37
112	172	130	69	76	25	38	93	48	67
7	84	63	28	42	55	83	123	64	55
0	61	72	40	54	36	43	73	44	60
0	51	74	47	70	23	23	47	59	103
25	74	90	55	86	16	25	31	58	100
53	107	109	61	99	16	45	24	40	55

Figure 2.3: The Matrix of Image Boundaries.

According to the above convolution calculation process, the dimension of the matrix is less than or equal to that of the original matrix after convolution operation. At the same time, after convolution processing, the element at the boundary and the surrounding non-boundary elements of the image matrix can be clearly found. The difference is further expanded. Therefore, after partial convolution layer processing, the special value of the boundary is obviously expressed, which is a serious effect, that is, the visualization of the boundary feature.

Since the first few layers can extract the boundary information, the convolution kernels of the first few layers can extract the boundary information (that is, expand the gap between the boundary and other regions at the matrix level), if so, the actual result is that as long as the following conditions are met

$$\sigma(x) \leqslant \sigma(s(x)) \tag{2.7}$$

where $\sigma(x)$ is the standard deviation of the initial matrix and $\sigma(s(x))$ is the standard deviation of the matrix obtained by the convolution. The extracted boundary information will conform to a certain law (on the matrix level), that is, some of the values are more obvious, and the values of the remaining parts are closer to each other and the difference between the previous part and the previous part is more obvious.

In this case, if above-mentioned convolution layer can work and feature extraction succeeds, according to the characteristics of the image boundary, after enough training to obtain the convolution layer in the experiment of different categories of targets, the layer can also have a certain effect, the effect may not be particularly good, but it is definitely not completely useless. In summary, based on the theory of migration learning, if the well-acquired convolution kernel is taken into an untrained network, this layer should be effective, which may be presented in the following aspects:

1. The trained convolution layer can be trained in a new network, and a good effect can be obtained, indicating that the convolution kernel is fixed and good, and also indirectly illustrates the latter pooled layer and fully connected layer. Parameters that interact with the convolution layer.

2. Trained convolutional layer used train in a new network, training time is shorter than direct re-training.

CHAPTER III

EXPERIMENT FOR THE FEATURE EXTRACTION

AlexNet is a historic breakthrough for CNN, and the VGG-16 generated after AlexNet has largely solved the problem of over fitting without too much tedious functionality. Therefore, this paper uses the VGG-16 network for experiments. From Flow Chart of the Convolution Neural Networks, it's easy to find that the CNN algorithm is not a normal defined iterative. And since convergence is a very mathematical vocabulary, it is impossible to evaluate the experimental results with the conventional definition of convergence. Base on that, in this paper, the experimental results are considered to be convergent when the recognition precision is high enough.

3.1 Introduction of the MSTAR Database

The experimental data is based on the measured SAR ground stationary target data published by the US-supported MSTAR program. The research on target recognition of SAR images is basically based on the data set. The sensor that collects the data set is a high-resolution spotlight synthetic aperture radar with a resolution of 0.3 m × 0.3 m. The collected data is preprocessed, and a slice image having a pixel size of 128×128 and containing various types of targets is extracted therefrom. This data is mostly a SAR slice image of a stationary vehicle, and contains target images acquired by various vehicle targets at various azimuth angles. A training set and test set recommended for the program are included in the data set. Targets of various categories also have different models, different models are slightly different but not much . The following image shows some images from the database.

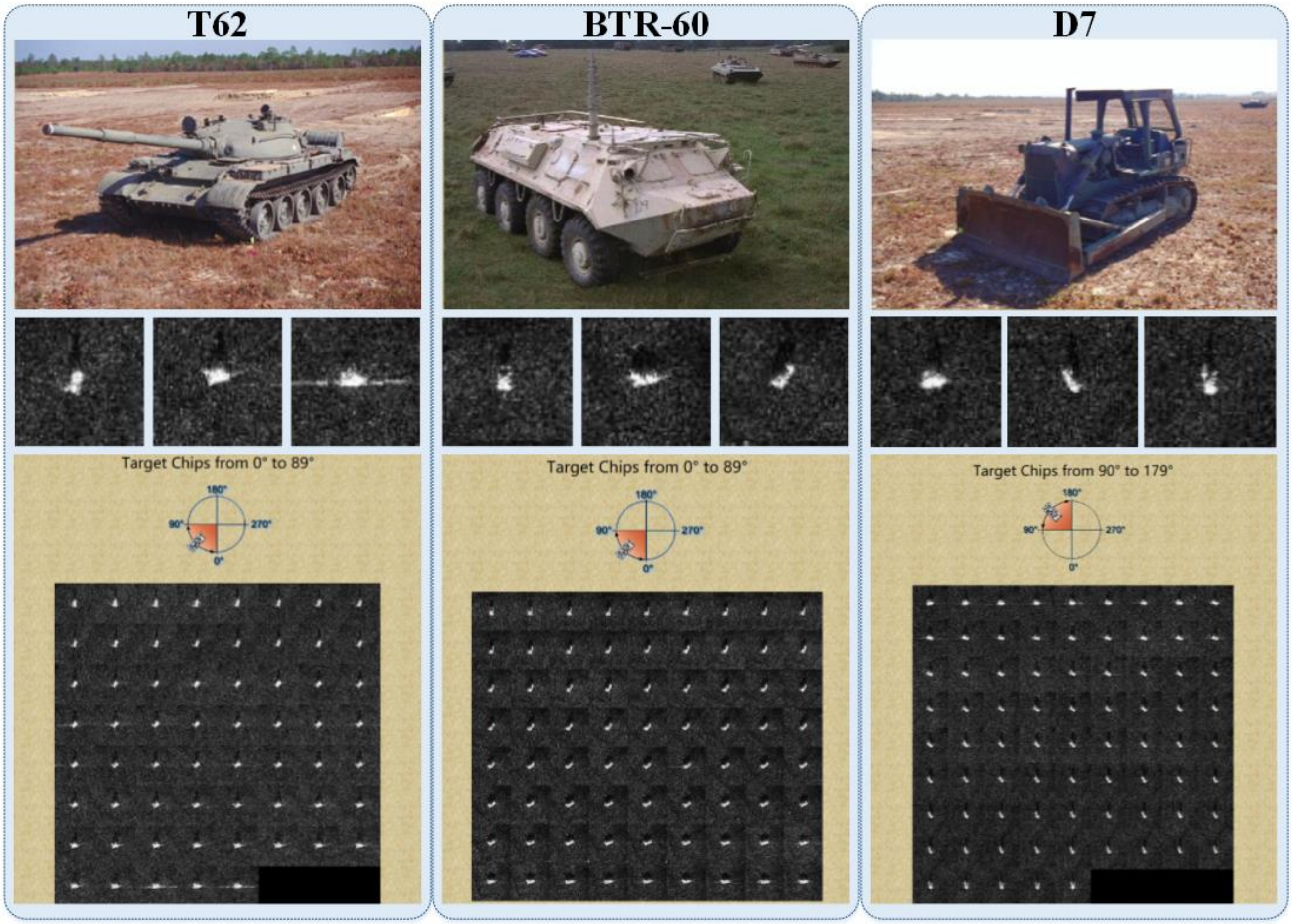


Figure 3.1: Military Objects in MSTAR Database.

3.2 Description of the Experimental Process

Since the MSTAR database is the most representative and versatile database for SAR image processing, this experiment will be performed using the MSTAR database. The process of the experiment is as follow:

1. Seven categories from the MSTAR database are selected for learning. After several iterations, the higher precision (Precision 1) is obtained, and are recorded along with the corresponding convolution layer parameters;

2. Based on the above experiment, the second learning is performed using the convolution layer parameters obtained by the first learning, and the obtained precision (Precision 2) corresponds to the precision when the convolution is unchanged. Compare and record

these two precision;

3. Four of the seven categories of data of the selected are randomly for learning, and the obtained highest precision (Precision 3) is recorded;

4. The remaining three categories are studied separately, recording the learning time (t1) and the highest precision (Precision 4) obtained in a certain period of time;

5. The last three types of data are used to fine-tune the network based on the training results of four categories, and the learning time (t2) and the highest precision (Precision 5) are recorded in a certain period of time.

3.3 Experimental Results

If the theory in Chapter 2 is correct, the Precision 2 should be greater than the Precision 1, and the Precision 4 should be less than the Precision 5, and at this time, the time t2 should be less than the time t1. The experimental results are shown in TABLE I, in which Experiment 1 to 5 correspond to the Precision 1 to 5 in Section 3.2.

Table 3.1: The Result of the Experiments

Experiments	1	2	3	4	5
Precision (%)	82.12	85.64	78.54	85.05	88.83
Learning Time (s)	-	-	-	2057	2049

3.4 Result Analysis

From the above experiments, the experimental results are basically in line with expectations, but have certain defects, though. For example, the precision of recognition can only reach 88%, and not fully accurate. At the same time, the time saved by secondary learning is not obvious. There are two main reasons for the above problems. The first one is that the image in the MSTAR Database is a radar image. If data is processed twice, their characteristics are not clear, so is the result extracted by the convolution layer. The second reason is that the results of the experiment are influenced by the number of experimental samples. MSTAR Database has only a few thousand images and the sample size is slightly less. At the same time, through the comparison of experimental results, it can be seen that in the experiment where the sample type is increased, that is, the sample number is increased, the experimental precision and time have been improved to some extent. Therefore, if the sample size is large enough, the experimental effect will be better. In summary, the experimental results are in line with expectations.

From the mathematics point of view, if a function eventually converges to a certain value, the function is said to be convergent. The experiments in this paper does not use a simple function, nor is it a simple iteration, so it is hardly to judge whether the result converges. Following the principle of convergence, it is assumed that the experimental data is unchanged and the recognition precision tends to an ideal precision (e.g. 100%). On the premise of this, the algorithm in the experiment converges, otherwise it is considered to be divergence.

CHAPTER IV

ANALYSIS OF EFFECT OF PADDING ON CONVOLUTION OPERATIONS

In previous chapters, the importance of the convolution layer was analyzed. The most important parts of the convolution layer are convolution kernel and padding, both of which play a decisive role in CNN network. In this chapter, selection of convolution kernel and effects of padding on the convolution layer will be discussed.

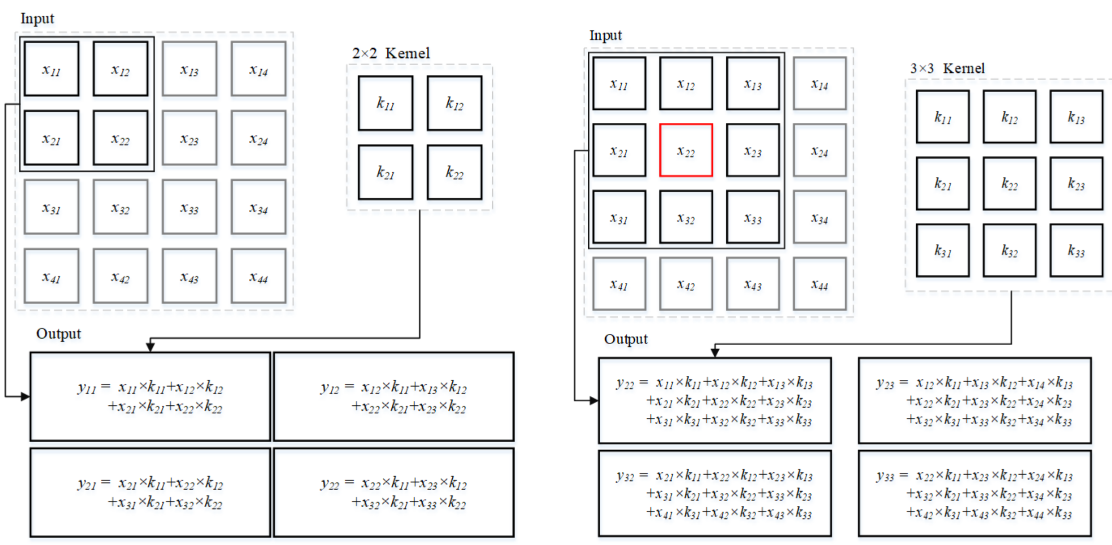
4.1 Selection Method of the Convolution Kernel

Since the development of CNN, people have invented various CNN models. Among these models , AlexNet, developed in 2012, is still a very typical one that has been studied extensively nowadays.

In AlexNet, the author Alex Krizhevsky used some very large convolution kernels, such as 5×5 or even 11×11 convolution kernels[1]. In the previous concept, people thought that the larger convolution kernel would provide the larger receptive field. In this situation, more picture information will receive, and the features extraction is better during the learning process. However, a large convolution kernel leads to an increase in the amount of data, which further leads to an increase in the amount of calculation. Because of computer technology limitations, the increase in the amount of calculation is not conducive to deepening the depth of the model. So in the Inception Network, the author proposed that the combination of two 3×3 convolution kernels is better than a 5×5 convolution kernel, and the parameter quantity is also reduced, so the 3×3 convolution kernel is later widely used in different kinds of models[3].

It can be found in a large amount of experimental data that people usually use odd-dimensional convolution kernels for feature extraction. The reason is that, comparing with even-dimensional (e.g. 2*2) convolution kernels, odd-dimensional (e.g. 3*3) convolution kernels have two unique advantages (as Figure 4.1 shows):

1. Odd-dimensional (e.g. 3*3) convolution kernels can ensure that the anchor point is just in the middle, and it is convenient to slide convolution with the module center as the standard;
2. When considering padding, the two sides of the image are still symmetrical if use the odd-dimensional (e.g. 3*3) convolution kernels.



((a)) 2*2 Convolution Kernel. ***((b))*** 3*3 Convolution Kernel.

Figure 4.1: The Difference between 2*2 Convolution Kernel and 3*3 Convolution Kernel.

4.2 Effect of Padding

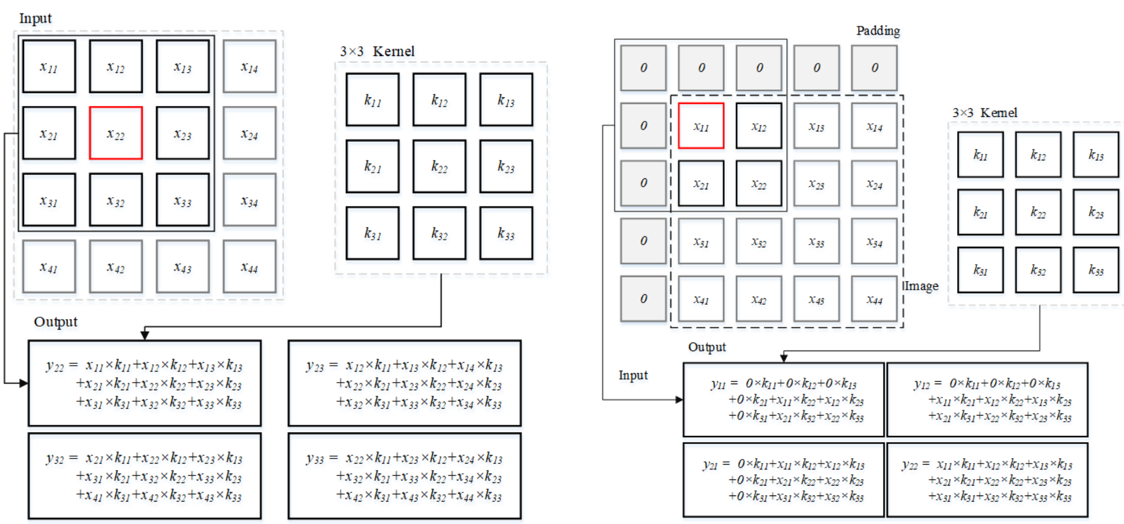
While doing the convolution, three problems will occur when only input images are considered:

1. The information at the edge of the input image will only be manipulated once by the convolution kernel, but the information in the middle of the image will be scanned many times, which will reduce the reference level of the boundary information to some extent;

2. Through the convolution operation, the size of the image (the dimension of the matrix) will drop rapidly, which will affect the further convolution operation, and may result in the extraction of some information that cannot be effectively extracted;

3. If the size of the input image (the dimension of the matrix) is different, the convolution operation cannot be consistent.

In order to solve the above problems, people add padding (usually 0) around the images.



((a)) 3*3 Convolution Kernel without Padding. *((b))* 3*3 Convolution Kernel with Padding.

Figure 4.2: The Difference between Convolution with Padding and Convolution without Padding

When some problems are solved with emergence of padding, new problems will be introduced:

1. In the process of computer processing, as the number of padding increases, the surrounding of the picture will become darker and darker;

2. Keeping the dimension of the matrix unchanged will make the amount of data larger and larger, which is not conducive to the actual operation;

3. If the size of the image itself is not large (the dimension of the matrix is not high), the added padding will account for a large percentage of the overall information, which will

result in the lack of valid information. For example, if the input matrix is a $n * n$matrix, we need to add $4n + 4$ 0 as the padding. The total number of information that has been read is $n(4 + n) + 4$, so the proportion of padding added is

$$p = \frac{4n + 4}{n(4 + n) + 4}.$$

When the input image is too small, the number of 0 may be more than the number of input elements.

When doing experiments, a certain amount of padding should be better than adding padding as much as possible. For example, if we have a $n * n$ image, if we add one "padding", the ideal proportion of the elements used for padding in the total pixels can be represented as

$$P_1 = \frac{\frac{5}{9} + \frac{1}{3} * (n - 2) * 4 * 2}{n * n} * p(d). \tag{4.1}$$

If we add two "padding", it can be

$$P_2 = \frac{[\frac{\frac{5}{9}+5+\frac{1}{3}*2}{9} * 4 + \frac{\frac{5}{9}+3+\frac{1}{3}*(2+1)}{9} * 4 * 2 + \frac{3+\frac{1}{3}*3}{9} * 4 * (n-2) + \frac{\frac{1}{3}*3}{9} * 4 * (n-2) + \frac{\frac{5}{9}+\frac{1}{3}*4}{9} * 4]}{n * n} * p(d) \tag{4.2}$$

where $p(d)$ is the percentage of drop out.

From the above, we can find that the more are added, the more the ideal proportion will be. This explains why we have to control the amount of padding added, not as much as possible. However, in the experiments implemented in Chapter 3 based on VGG-16 network, the experimental precision is not significantly different when different numbers of padding are added in the convolution layers. The fundamental reason is that most of the objects in the database are centered in the middle of the images. In this case, the factors affecting the boundary cannot be adequately reflected in the experimental results.

4.3 Description of the Experimental Process

In order to solve this problem, I divided and reorganize the image. The actual effect is as follows.

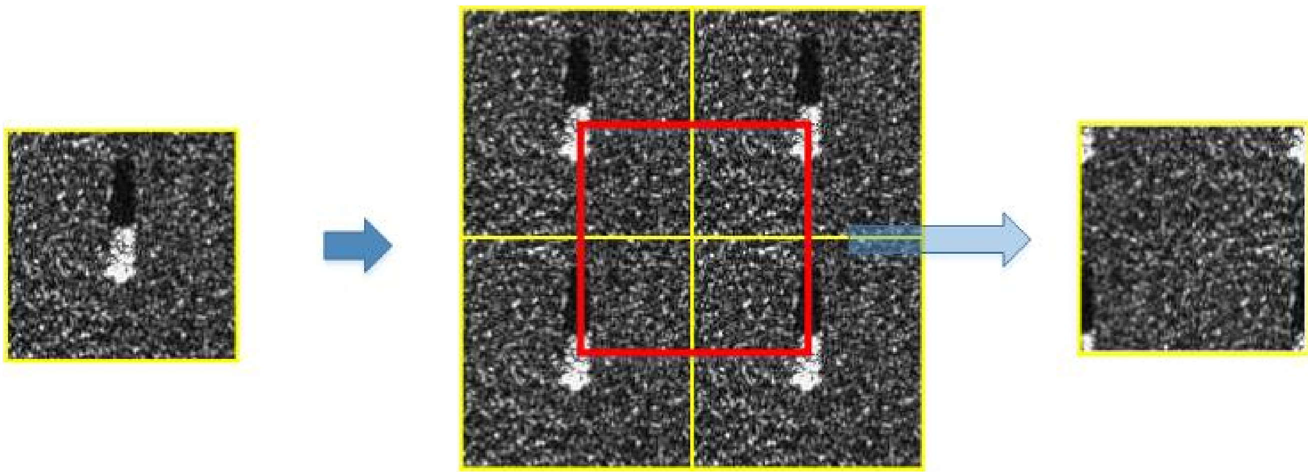


Figure 4.3: Image Segmentation and Reorganization.

These images will be used in the experiment. In the case of keeping the network structure unchanged, choose to add padding in different convolution layers to conduct experiments. If the above theory is correct, the precision will be highest when adding 8-10 layers of padding.

4.4 Experimental Results and Analysis

Table 4.1: The Result of the Experiments

The number of padding	4	5	6	7	8	9	10	11	12	13
Time(s)	5371	5462	5731	5848	6259	5774	5777	6722	5987	6545
Precision(%)	40.90	49.91	52.79	76.14	74.75	81.05	81.56	79.22	79.95	77.47

As the experimental results show, padding has a great influence on the image boundary, which will affect the extraction of information, and thus affect the precision of the experimental results. The experimental results are in line with expectations. So in following experiments, the number of suitable padding needs to be considered.

The precision of the experimental results is not high because the pictures whose characteristics are not obvious are further blurred by the second processing. If there is an ideal database, the experimental precision will be in line with expectations.

CHAPTER V

CONCLUSIONS

In the previous two chapters, we experimented with the convolution layer and the corresponding padding, and obtained the expected experimental results. However, for some reasons, the experiment still has certain limitations. In this chapter, we will summarize the results, explain the shortcomings of the experiment and show the future direction of improvement.

5.1 Conclusions

Convergence is a very mathematical vocabulary, and CNN is not a pure iterative process, so it is impossible to describe the experimental results of CNN with the normal defined convergence. Therefore, in this paper, if the recognition precision is high enough, the experimental results are considered to be convergent.

As long as the convolution kernel whose calculated element value is larger than the standard deviation of the original matrix region is satisfied, the (boundary) feature of the image can be well extracted. At the same time, according to the calculation method of convolution, training the trained convolution kernel to the new network can still play a certain role, and to some extent, strengthen the training effect, and reduce the time required for training.

Padding also plays a big role in the feature extraction part, and the suitable number of padding will make the experimental results more ideal. We need to measure the number of elements in the input matrix and the number of elements in the padding added to balance the amount of data. On this basis, we can better play the role of padding.

In summary, if exclude the effects of the image itself and the sample size, and the experimental results are in accordance with expectations, that is, the recognition precision satisfies the requirements. The experiment is successful, and the effectiveness of the convolution is proved.

5.2 Limitations and Future Work

This study has two potential limitations. The first is that the number of samples required for the experiment is limited and may not be adequately tested, resulting in omissions in the results. The second point is that the structure of the algorithm is not very complicated and still has the possibility of improvement.

In my future study, I can increase the amount of data on the basis of artificial processing (such as add noise, offset, etc.), and try to create a new network independently, so that the results of the experiment are more targeted. Also, I will try to find an ideal loss function which can decide the padding's amount. On the basis of the above, the proposed theory is used to further optimize the network structure, so that the experimental results meet the needs of the application.

REFERENCES

[1] I. S. A. Krizhevsky and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, Advances in Neural Information Processing Systems, (2012).

[2] Y. Bengio, *Learning deep architectures for ai*, Foundations and Trends® in Machine Learning, 2 (2009), pp. 1–127.

[3] S. I. J. S. Z. W. C. Szegedy, V. Vanhoucke, *Rethinking the inception architecture for computer vision*, The IEEE Conference on Computer Vision and Pattern Recognition, (2016).

[4] G. E. H. David E. Rumelhart and R. J. Williams, *Learning representations by back-propagating errors*, International Journal of Science, (1986), pp. 318–362.

[5] M. D. Z. Fergus, *Visualizing and Understanding Convolutional Networks*, vol. 8689, September 2014.

[6] R. R. S. G. E. Hinton, *Reducing the dimensionality of data with neural networks*, Science, 313 (2006), pp. 504–507.

[7] S. R. K. He, X. Zhang and J. Sun, *Deep residual learning for image recognition*, The IEEE Conference on Computer Vision and Pattern Recognition, (2016), pp. 770–778.

[8] D. G. Lowe, *Object recognition from local scale-invariant features*, Proceedings of the International Conference on Computer Vision, 2 (1999), p. 1150–1157.

[9] J. Schmidhuber, *Deep learning in neural networks: An overview*, Science Direct, 61 (2015), pp. 85–117.

[10] B. J. M. L. Y. Säckinger Eduard, Boser Bernhard E and J. L. D, *Application of the anna neural network chip to high-speed character recognition*, IEEE Transactions on Neural Networks, 3 (1992), p. 498–505.

[11] J. N. T Sihvo, *Row-column decomposition based 2d transform optimization on subword parallel processors*, International Symposium on Signals, (2005), pp. 1–127.

APPENDIX A

APPENDIX A

CODE FOR IMAGE SEGMENTATION AND REORGANIZATION.

img=[];

PathName = ’PathName’;

cd(PathName)

filelist = dir(PathName);

for i = 3:length(filelist)

temp$_name = filelist(i).name$;

temp$_img = imread(temp_name)$;

img = cat(3, img, temp$_img)$;

end

[m,n,]=size(img);

s=ceil(m/2);

t=ceil(n/2);

NewImg = [img(s+1:m,t+1:n,:),img(s+1:m,1:t,:);img(1:s,t+1:n,:),img(1:s,1:t,:)]; x$_train = permute(NewImg, [312])$;

label=uint8([1*ones(1,195) 2*ones(1,196) 3*ones(1,196) 4*ones(1,196) 5*ones(1,196) 6*ones(1,195) 7*ones(1,191)]);

y$_train = permute(label, [21])$;

save(’x$_train.mat','x_train')$

save(’y$_train.mat','y_train')$

BIOGRAPHICAL SKETCH

Ligang Zou was born in Heilongjiang, China on January 17, 1995. He completed his master with a major in Statistics in August 2019 from University of Texas Rio Grande Valley. He graduated from Northeast Forestry University in June 2017 with a Bachelor of Science.

During his graduate education, he worked in Dr. Zhijun Qiao's group. At the same time, he was hired as a Teaching Assistant by SMSS department of UTRGV.

Ligang concentrated on applied math area and attend the SPIE conference two times with his work. With the graduate degree, he plans to pursue a second master degree in different major and become more professional in this interesting field. He can be contacted via email ligang.zou01@utrgv.edu.