Classification and Detection with Convolutional Neural Networks

Shanshan Jiang
sjiang97@gatech.edu

1. Introduction

In recent years, the convolutional neural network (CNN) has achieved great success in many computer vision tasks. In my project, I also use CNN to find the digital number from the images. It uses the MSER algorithm to extract patches, and it borrows the structure of VGG-16 with some changes, which are using the residual block to replace the Conv layer in the VGG-16 structure — then using the Non-maxima suppression to detect the object. It is using digital dataset SVHN combined with a non-digital dataset which downloaded from the Flick.

2. Previous Work

In this section, I will talk about the existing methods published in recent research.

- In the paper "Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree", which published in 2016, has error result around 1.69%. They improve the depth of neural networks by promoting pooled operations that play a central role in the current architecture. They have shown experimentally that their proposed pooling operation provides invariant properties and sets the latest technology on several widely used baseline datasets, which are also easy to implement and can be applied to a variety of deep neural network architectures. These benefits lead to a slight increase in computational overhead during training and a very modest increase in the number of model parameters.
- In the paper "Competitive Multi-scale Convolution," It introduces a new competitive deep convolutional neural network (ConvNet) module which can promote a set of multi-scale convolution filters. It has two goals: 1) To prevent the filter from working together and to allow the formation of multiple subnets within the same model. 2) The Maxout unit reduces the dimensions of the multi-scale filter output. These have been shown to contribute to the training of complex learning problems.(Liao…,2015)
- In the paper "Recursive convolution neural Network for object recognition", it proposes a follow-up of object recognition, CNN (RCNN), which combines cyclic connections into each convolution layer. We know that the activity of the RCNN unit is developing, although it always has a static input so that the adjacent units will regulate their activities with each other. This approach enhances the ability of the model to integrate contextual information, which is critical for object recognition. Besides, the method obtained the result of 1.77% error rate.(Liang…,2015)

3. Algorithms

In this section, I will talk about how the algorithm worked on my project. It is showing as in figure 1 it has a dataset as input in the beginning, then extracting the patches to be train. Also, it uses CNN to classify the input as different patterns. For each step, it accepts an input and produces an output until all the dataset has completed. Through the training result, it decides the number string, then outputs the exact result. In the subsection, I will give more details.

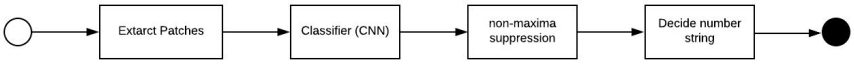

Figure 1. Structure

3.1 Extract Patches

In this section, I will talk more about Extract Patches. When users input some images, Usually, the process is to scan each pyramid to find each pattern, then putting them in the scanner window. But which takes much time. So in my project, I used an algorithm called MSER to improve efficiency.

- Maximally Stable Extremal Regions (MSER):
  MSER is a feature detector, which used as a method of blob detection in an image, which is the way to extract a full number of corresponding image elements. It can help with comprehensive baseline matching, and it can also lead to better stereo matching and object recognition.(Maximally…,2019)

3.2 Classifier(Convolution Neural Networks)

- Convolution neural networks(CNN):
  They are very similar to ordinary neural networks, which consist of neurons, each receiving some input, performing dot product and optionally following nonlinearity. However, the input to the ConvNet architecture is an image that allows specific properties to be encoded into the architecture. Then in the network, it enables forward functionality to be implemented more effectively and significantly reduces the number of parameters.The convolution neural networks consist of several different layers. Typically, they include the input layer, the hidden layer, and the output layer. They are more suitable for object detection. (Convolutional…,2019)

- VGG-16 Structure:
  I borrowed the structures from vgg16 showing figure 2. However, vgg16 cannot be used directly in the project. Because vgg16 has 1000 classes, but the project only has 11 classes. So there were some changes has been made to the project according to vgg16 (VGG16..,2018). It changes the last connection layer from 1000 to 11. Moreover, there are two sets of training methods: one is to use Pre-train weight, the other is not to Pre-train weight.

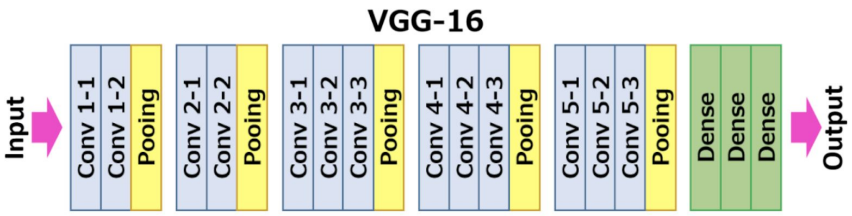

Figure 2. VGG-16 Structure

- **The proposed Network Structure:**
  By looking at figure 2, we can find there are many Conv layer. So what I made in my project is I changed the Conv layer to residual block showing figure 3. The research is showing that the ResNet outperforms than others because it is easy to push the residuals to zero (F(x) = 0). In a simple language, using a bunch of nonlinear CNN layers as functions, it is able to directly derive solutions like F (x) = 0 instead of F (x) = x.(Jay…,2018)

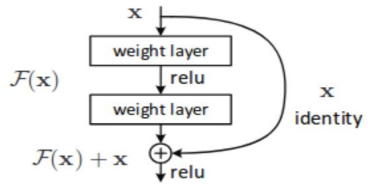

Figure 2. Residual learning: a building block.

Residual block

Figure 3. Residual Block

3.3 Non-maxima Suppression

- NMS:
  NMS is used to ensure that specific objects are recognized only once in object detection. For example, when a user recognizes an object, but multiple bounding boxes around the same object, the user tries to keep only one border without deleting the candidate object for the different object.
- Suppression in my project:
  In the project, the suppression process base on two aspects: First, it has to make sure that multiple box point at the same object; Second, it also has to make sure that the boxes are close enough, which is by calculating the average.

4. Results

4.1 Network Training

- Dataset:
  I am using dataset SVHN. And I used to form 1 Dataset in SVHN, and the dataset in form 1 included training, testing, and extra, and I took the extra data into the training data. However, because the data in SVHN are digital numbers from 0-9, I need to add some non-numeric data. I downloaded some data labeled street, nature, city, people and house from the flick. Then I cropped them

randomly and ended up generating 60,000 non-numeric datasets. At the end of the process, there are 2570,000 images produced.

- Data Augmentation:
  The project is asking to be robust in Scale invariance, Location invariance, Font invariance, Pose invariance, Lighting invariance, and Noise invariance. The MSER has already helped to fix the location invariance. Also, the dataset has the font invariance. For other invariance, I have already done some robust code to fix such flexible change.
- Training Details:
  GPU: NVIDIA, TITAN;
  Learning rate: 0.0001;
  Batch site: 1024;
  Epochs: 800;

4.2 Comparison

- Proposed Network Structure:
  I got some comparison result between Vgg-16 and proposed structure showing in Figure 4,5 below. Basically, the proposed structure result is better than the result of Vgg-16. By looking at Figure 4, there are two graphs. The left one is showing the accuracy of the proposed network structure. The x-axis is the Epochs, and the y-axis is the Accuracy. It was evident that the result of train is better than the result of validation. Moreover, the validation curves look very unstable. The right one is showing the loss of the proposed network structure. The x-axis is the Epochs, and the y-axis is the Loss, which is the categorical cross entropy function. It was also evident that the result of the train is better than the result of validation. So I got the training result of accuracy is around 98% and the test result of accuracy is around 97.4%, which I did not show in the report.

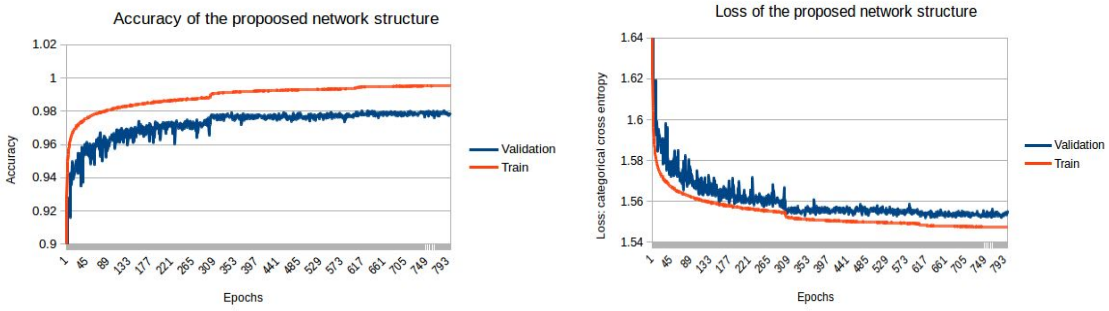

Figure 4. Left: Accuracy; Right:Loss

- VGG-16:
  Another comparison in this section is between the accuracy of vgg16 weight with pretrained and weight without pretrained showing in figure 5. I can tell that the result with pre-trained is better than the result without pre-trained. The accuracy with pre-trained weights is around 99%, and its test result is around 96.8%. The accuracy without pre-trained weights is around 98%, and its test result is around 96.1%, which is showing in the program.

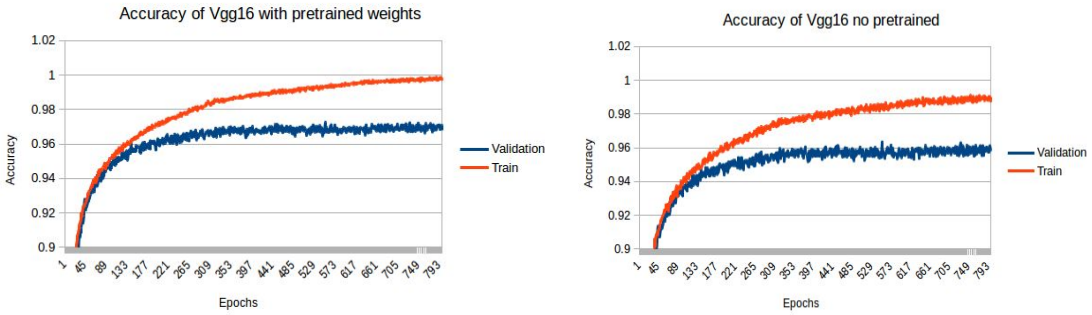

Figure 5. Left: Result of Pre-trained Weights; Right:Result of without Pre-trained Weights

4.3 Discussion

In the results, there are rare cases where something that is not a number is recognized as a number, as shown in the figure 5. That is because some letters look particularly like numbers, such as g likes 9. There are too many non-numeric things in our dataset. To solve this problem, I think there are three ways: First, we need to increase the number of negative cases; Second, it is necessary to learn the pattern through the other combine numbers in the entire digital mix. And third, Adding to datasets to train can improve its ability to identify digital shapes.

Figure 5. Failure Fully Recognize Pic.

Figure 6. Fully Recognize Pic.

5. Conclusion

Looking back at My Network training section, I processed the dataset and finally produced 2.57 million images. And I made the code more robust in order to do Data Augmentation like Lighting invariance and so on. For the training data, by comparison the results between pre-trained weights and trained weights in VGG-16, I found that the result of pre-trained weight was significantly better than the trained. In the comparison of the Proposed Network Structure, the results of the train are better than those of the validation. For all the comparison, The results of the test data have the same situation.

Through the paper "Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree," we know the result of the state of the art, which using some new pooling function to improve its best result. For my work, I am using the max pooling function, and it is learnable pooling. In order to improve my result, it may change the structure to some deeper to train the dataset. Also, it may make use of some pre-trained network.

6. Reference

1. Karpathy, A. (n.d.). CS231n Convolutional Neural Networks for Visual Recognition. Retrieved April 15, 2019, from http://cs231n.github.io/convolutional-networks/

2. Convolutional neural network. (2019, April 09). Retrieved April 16, 2019, from https://en.wikipedia.org/wiki/Convolutional_neural_network

3. Lee, C. Y., Gallagher, P. W., & Tu, Z. (2016, May). Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. In *Artificial Intelligence and Statistics* (pp. 464-472).

4. Maximally stable extremal regions. (2019, February 19). Retrieved April 19, 2019, from https://en.wikipedia.org/wiki/Maximally_stable_extremal_regions

5. VGG16 - Convolutional Network for Classification and Detection. (2018, November 21). Retrieved April 19, 2019, from https://neurohive.io/en/popular-networks/vgg16/

6. Jay, P., & Jay, P. (2018, February 07). Understanding and Implementing Architectures of ResNet and ResNeXt for state-of-the-art Image... Retrieved April 19, 2019, from https://medium.com/@14prakash/understanding-and-implementing-architectures-of-resnet-and-resnext-for-state-of-the-art-image-cf51669e1624

7. Liao, Z., & Carneiro, G. (2015). Competitive multi-scale convolution. *arXiv preprint arXiv:1511.05635*.

8. Liang, M., & Hu, X. (2015). Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3367-3375).