Project 1.1 & Project 1.2

Hui Wang, s210331, Jianting Liu, s210141, Jiahao Li, s210313

Project 1.1

1.1 Introducion

In this project, we design and train a CNN network to classify images into two classes: hotdog or not hotdog. Firstly a CNN network with parameters set by us is constructed to train the model and see the performance of test accuracy. Then data augmentation are implemented to expand trainset and help with the improvement of accuracy. Finally, transfer learning is applied to increase the result performance and show how big the difference is among selected models and the initial CNN network.

1.2 Neural Network Architecture

There are three layers in convolution network and three layers in fully-connected network (figure 1.1). A 3 × 3 kernel with different channels in different layers is used as well as Maxpool to capture multiple features of image.

1.3 Efficient Optimizer and Data augmentation

To compare the performance with different optimizers, we train our model with SGD and Adam optimizer. The performance is shown in (figure 1.2). As a result, Adam optimizer has a significant improvement on accuracy.

In order to improve performance, we also try some other ways. For example, we implement Data augmentation to expand trainset and reduce probability of over-fitting. Through figure 1.3 we can find that, after implementing data augmentation, the accuracy of train is reduced because the difficulty of learning is increased for the model, thereby improving the overall performance of the model

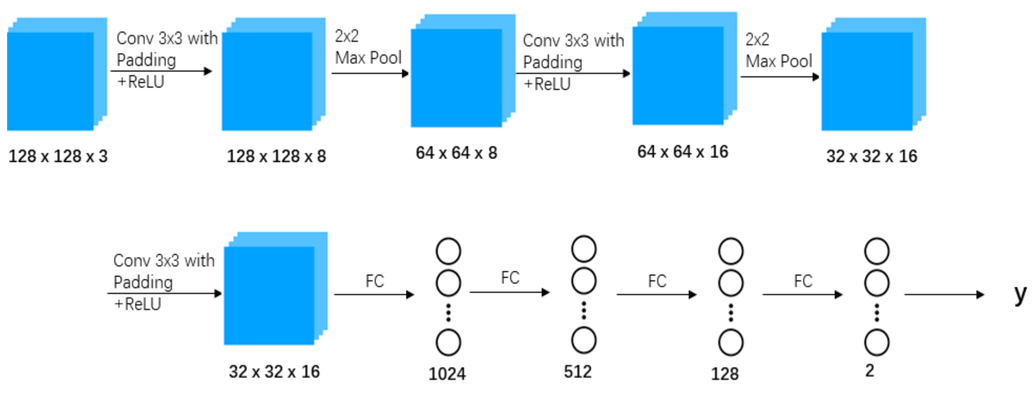

Figure 1.1 Neural Network Architecture

1.4 Transfer Learning

We select two different models(ResNet and VGG) to see how dramatical improvement they have on the accuracy and which model performs better.

Table 1.1 Performance with different models

Model	CNN	ResNet 18	VGG 16
Accuracy	77.88%	86.45%	91.14

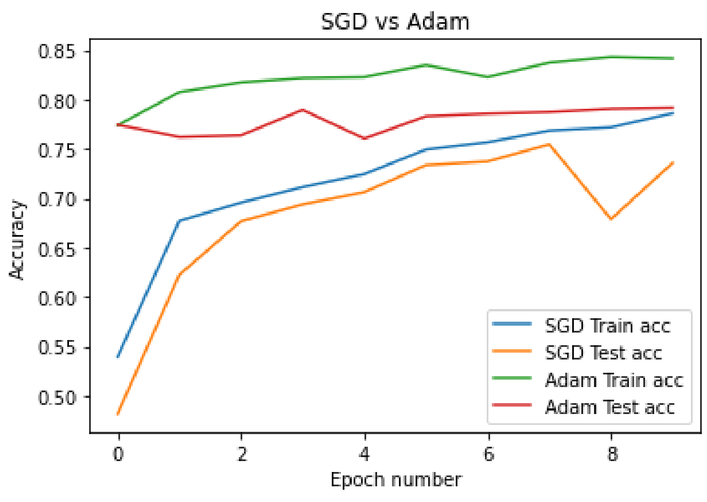

Figure 1.2 Comparision between SGD and Adam

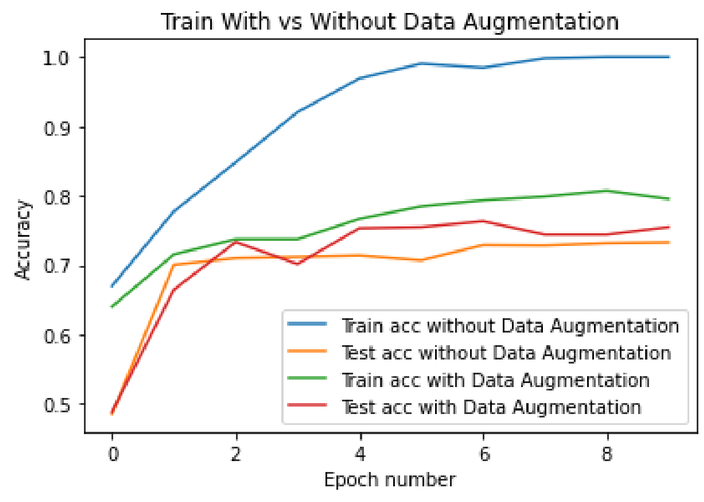

Figure. 1.3 Comparision in data augmentation

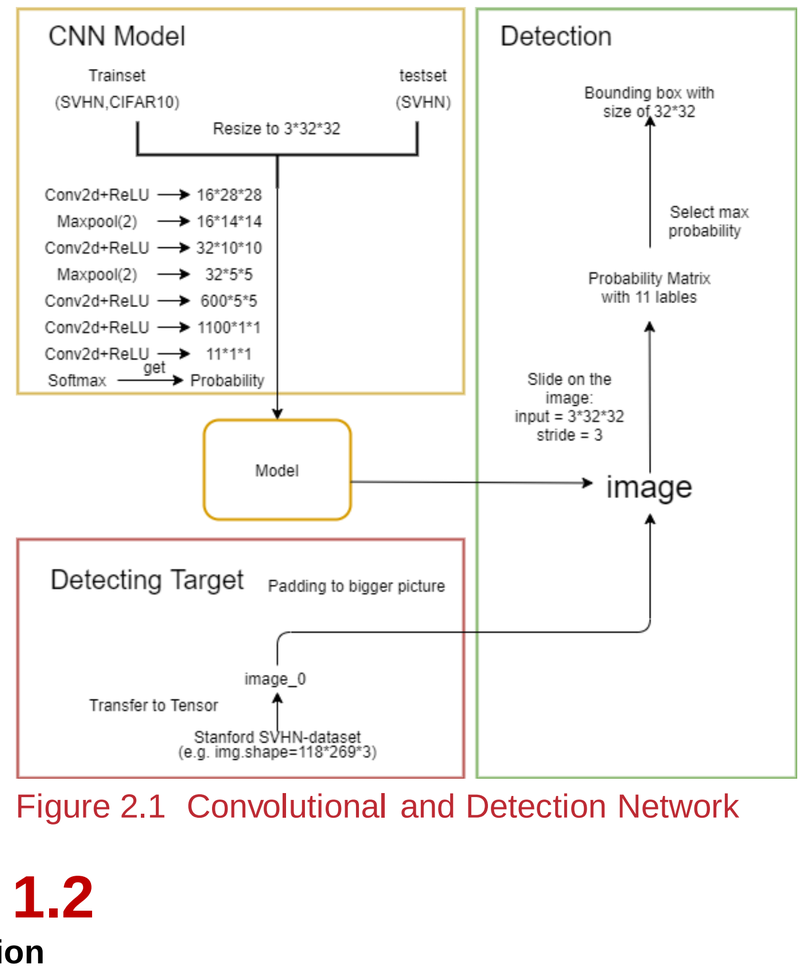

Figure 2.1 Convolutional and Detection Network

Project 1.2

2.1 Introduction

In this project, we need to use SVHN datasets to train a model that can be implemented to classify the number in an image and another goal is to detect and localize the position of digit. CIFAR10 is imported as trainset with no digits. Images of 32*32 size in SVHN is used as trainset while full-sized images is served as testset.

2.2 Neural Network Architecture

Specially, only Convolutional Network is used instead of using Full-Connected Network. The size of train image can be converted from 32 × 32 × 3 to 1 × 1 × 11 by choosing suitable kernel, which size is 5 × 5 and 1 × 1, and doing Maxpool. The network architecture is depicted in figure 2.1

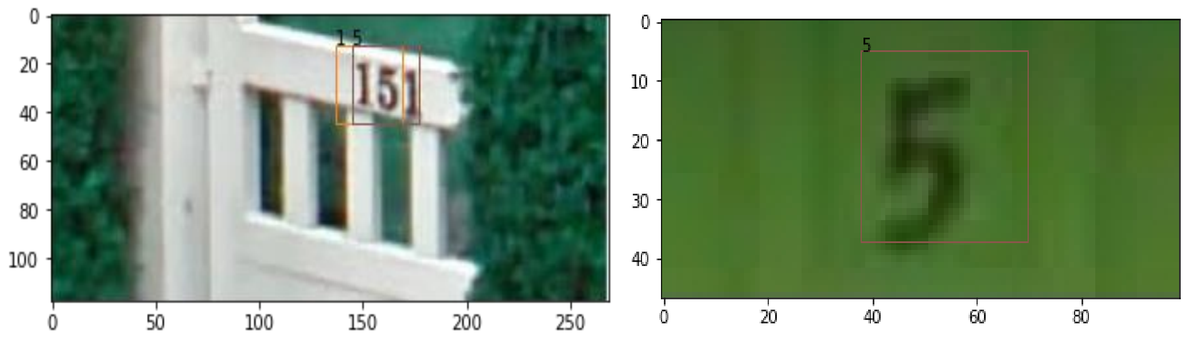

Figure 2.2 Bounding Box

2.3 Digit Detection and Bounding Box

Full-sized image from testset is processed by padding to be a fixed-size image(such as 128 × 128 size). The fix-sized image is divided into serveral sub-images with size 32 × 32. Then the trained model is implemented to every sub-image to get a 11 × h × w tensor, the value in which represents the probability of different digits. In the other word, the information of every sub-image is stored in a 1 × 1 × 11 tensor and the 11 values represents the probablity of different numbers in the sub-image. The maximum value and its position in tensor is extracted. The digit that maximum value represents is considered to exist in the sub-image. According to the position, we can get where the sub-image is in the fixed-sized image. If we use(x,y) to depict the position of pixel in a channel, then the range of corresponding area in orgin image is[(4x:4x+31),(4y:4y+31)]. The detection results are shown in figure 2.2.