CS 6476 - Classification and Detection with Convolutional Neural Networks

Yihuan Su

ysu312@gatech.edu

1 Introduction

The goal of this project is to develop a system that's capable of detecting digits in a given image. A use case for the system is to detect street number of a home address in an image. To achieve this I first apply Maximally Stable Extremal Regions (MSER) to extract regions of interest (ROI), then Convolutional Neural Network (CNN) is used to recognize the digits in the ROI. In this report, I first introduce some works that are related to the topic. Then I describe the methods I used to develop the system. In the end, some results are demonstrated.

The CNN model is developed based off pretrained weights of VGG16 [2]. The Street View House Numbers (SVHN) Dataset (ufldl.stanford.edu/housenumbers/) is used for training.

2 Related Work

CNN is a type of neural network model designed for data with a grid-like topology that are known. Unlike regular neural networks which use general matrix multiplication as layers, CNN use convolution in at least one of the layers. This allow CNN to deal with large images with high efficiency. Pooling layers are often used in CNNs when dealing with images. A pooling layer facilitates the summarization of the activations of adjacent filters with a a single response [1].

A lot of work have been done on detecting digits in images utilizing CNNs. Notably, Goodfellow et al [2] proposed a multi-digit number recognition system for SVHN which unifies localization, segmentation, and recoginition steps using a Deep CNN. Another prominent method that are applicable to digits detection is You Only Look Once (YOLO) which was proposed by Redmon et al [3]. It use a single neural network to predict bounding boxes and class probabilities from an image in just one step.

3 Method

For this project, a two step approach is adopted to solve the problem. First, I utilize MSER to extract regions of interest. Second, a CNNs classifier with 11 classes, which are 'not a digit' and digits 0 to 9, is applied to detect if there is a digit in the region and what's the digit.

3.1 Extract Regions of Interest

Maximally Stable Extremal Regions (MSER) is a blob detection method that was first proposed by Matas et al [4]. Originally, the goal was to find corresponding image elements from two images with different view. With MSER algorithm, a series of threshholds is applied to an image. The pixels below the threshold are white and above or equal to the threshhold are back. This creates a sequence of black and white images. The set of of all connected components in the sequence becomes the set of extermal regions.

When applying MSER to images to extract regions of interests for this project, it returns a lot of regions. Therefore, it is necessary to prune the regions. Based on ideas proposed by Islam et al [5] and my owned intuition, I apply the following rules to prune the regions extracted by MSER. First, it's intuitive to make the assumption that one single digit can't occupy more than 30% of the image, at the same time, it can't be too small to read. So all the regions that occupy more than 30% or less than 0.1% of the image are removed. Second, with the assumption that the bounding box of a digit should be an upright rectangle that can't be too narrow, all the regions of which the aspect ratio (width/height) of bounding boxes that are less than 0.2 or great than 1 are move. After these pruning steps, I applied Non-maximum Suppression to remove regions that overlap with other regions so each resulted region is unique. A reasonable small number of regions are left after these steps.

3.2 Convolutional Neural Network

The training set and test set of cropped digits data from The Street View House Numbers (SVHN) Dataset (ufldl.stanford.edu/housenumbers/) are obtain for training the CNN model. In addition, 32x32 no digits images generated using random online images are appended to the training and testing set.

Transfer learning is to exploit what has been learned in one setting, in order to improve generalization is a different scenario [1]. Transfer learning is often applicable for image recognition tasks, because regardless of the objective of the model, the basic features of images, like lines and shapes, are the same ac across different images. For this project, I adopted pretrained weights and layers off VGG16 [6] to train my CNN model. Based on VGG16 without the dense layers, I defined two dense layers each with 4096 nodes and relu activation function, as well as an output layer with softmax activation to classify the 11 different class.

Training a neural network model is to minimize a loss function through forward and back propagation. For this model, category cross entropy loss, which is suitable for multi-class classification problems, is used. category cross entropy loss function is written as:

$$Loss = -\sum_{i=1}^{n} y_i log\hat{y}_i \tag{1}$$

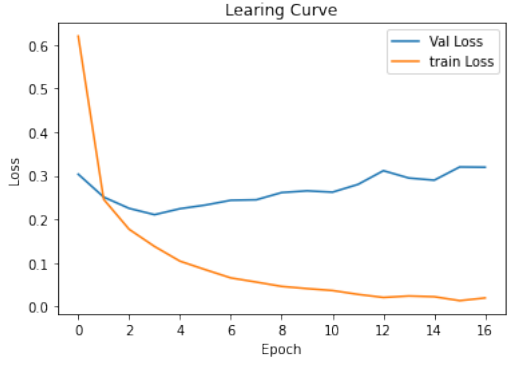
where n is the number of classes, $\hat{y}_i$ is the model output for ith class, and y_i is the corresponding target value.

Gradient descent is the most commonly used algorithm for optimizing loss functions in Machine Learning. The idea is to repeatedly take small steps with respect to a learning rate along the gradient of the function at a point in order to find the global minimum, given loss functions are convex. When there is a large set of sample, the training process may be accelerated through the adoption stochastic gradient descent (SGD). Instead of following the gradient of an entire training set downhill, SGD uses randomly selected minibatches. SGD and its variants are the most commonly used optimization algorithms for deep learning, due to the efficiency benefit over gradient descent. At the same time, it is possible to get an unbiased estimate of gradient by taking the average gradient an a minibatch [1].

Adam is one of the variant of SGD algorithm. It is what I used for optimizing the CNN model for this project. Adam is a adaptive learning rate optimization algorithm. In comparison to SGD of which the learning rate is constant through out the training process, the learning rate of Adam algorithm is adjusted based on the first order moments and the second order moments during training. This allows Adam to converge faster than SGD. Adam also often converge better, i.e. get closer to local minima, than SGD [1].

I also applied early stopping during training in order to prevent overfitting. Early stopping is likely the most commonly used regularization method in deep learning. The idea is to monitor validation loss during training, and stop training when validation loss stops to decrease or even start to increase. Fig 1 shows that while training loss continue to decrease as the number of epoches increases, validation loss start to increase after epoch 4. So it's good idea to stop the training process at that point.

Fig. 1. Training and Validation Loss by Epoch



Due to limited computational resources and time, I was unable to perform an exhaustive hyper-parameter tuning. The table below shows experimentation with lean ring rate and batch size. Based on validation accuracy, learning rate=0.0001 and batch size=32 are selected.

Learning Rate	Batch Size	Val Accuracy	Test Accuracy
1^{-4}	32	0.9483	0.9538
1^{-4}	64	0.9470	0.9531
1^{-4}	128	0.9475	0.9530
1^{-5}	32	0.9331	0.9399
1^{-5}	64	0.9326	0.9388
1^{-5}	128	0.9213	0.9285

4 Experiment

Fig. 2. The System is Capable of Detecting Digits in Difference Scenarios

When applying the system to images, it often works very well. Images in Fig 2 show the model is able to correctly detect digits at different scales, different orientations, different location, and different lighting conditions.

However, there are also situations where the system doesn't work as well. Fig 3 shows two images at which the system failed to detect the digits correctly. In the first image, the CNN model failed to recognize the '1' correctly, because it looks very much like a '7'. For the second image, the resolution of the image is very low, and the digits are very small. The digits are unrecognizable even for human eyes. The system failed to extract the regions where the digits resides. It also falsely considered regions that resemble number as digits.

Fig. 3. The System Fails Sometimes

5 Conclusion

In the project, I successfully developed an digits detection system by first applying MSER to extract ROI, then using a CNN model to recognize if there is a digit in a ROI and what digit it is. The system is capable of detecting digits correctly from images in various scenarios. But it's not perfect. It can fail in other situations.

References

1. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning
2. Ian J. Goodfellow, Yaroslav Bulatov, J.I.S.A.V.S.: Multi-digit number recognition from street view imagery using deep convolutional neural networks (2014) arXiv:1312.6082 [cs.CV].
3. Joseph Redmon, Santosh Divvala, R.G.A.F.: You only look once: Unified, real-time object detection (2015) arXiv:1506.02640 [cs.CV].
4. J. Matas, O. Chum, M.U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. Proc. of British Machine Vision Conference (2002) 384–396
5. Rashedul Islam, R.I., Talukder, K.: An enhanced mser pruning algorithm for detection and localization of bangla texts from scene images. The International Arab Journal of Information Technology **17**(3) (2020) 375–385
6. Karen Simonyan, A.Z.: Very deep convolutional networks for large-scale image recognition (2014) arXiv:1409.1556 [cs.CV].