Street Number Recognition
Group 67

Research School of Computer Science,

Australian National University

Abstract: Recently, numerical recognition is useful in various areas. In this project, we aim to detect and recognize street numbers in real world. We split the Street View House Number dataset to training and testing datasets for data processing. Maximally stable extremal regions method and a convolution neural network detector are used to detect the images including numbers. Then we use the Non-Maximum-Suppression method and a convolution neural network recognizer to recognize specific numbers. The result shows that the accuracy for the detector is 90%, and the accuracy for recognizer is 88%. This basic working flow provides individually digits spot and recognition and shows its performance in real world challenges

1.Introduction

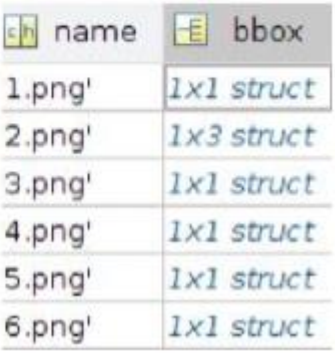
1.1 Background and Motivation

Nowadays numerical recognition is widely used in our life. Numerical recognition brings convenience to people and based on numerical recognition, many technologies such as drone's delivery come true. In 2013, Google makes a breakthrough in numerical recognition. Street View House Numbers (SVHN) is a data set used in Google's method for the recognition of numbers in real-world images. All images in SVHN comes from house numbers in Google street view. Data preprocessing is minimally required in SVHN, so it is a suitable data set for developing machine learning. Although SVHN had been widely used as training and testing data set, dealing with this huge amount (over 600,000 images) data can be a challenge.

Recently, with the fast development of deep learning technique, a great number of solutions to image understanding tasks have been proposed., Alex and Ilya developed a large, deep convolutional neural network for classification [1]. This neural network can classify 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. Huang et al. introduce the Dense Convolutional Network (Dense Net) [2], which connects each layer to every other layer in a feedforward fashion. And some other similar works such as RCNN [3], given by Liang and Hu.

In this paper, we introduce our program that used SVHN as training and test data set and the program can detect house number in the real world and determine what numbers are. We only used 20% images in SVHN dataset, therefore, the difficulty is reduced.

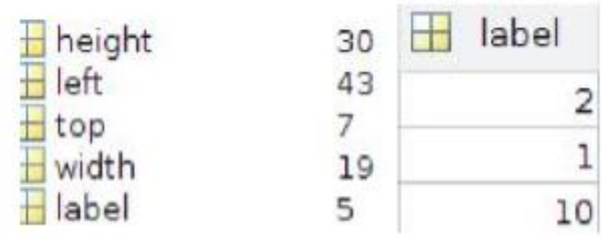
1.2 Dataset



name	bbox
1.png'	1x1 struct
2.png'	1x3 struct
3.png'	1x1 struct
4.png'	1x1 struct
5.png'	1x1 struct
6.png'	1x1 struct

Figure 1.2.1 data struct

We used SVHN [5] dataset. The dataset includes . png format colour house-number images and information of bounding box in. mat format files. All house-number images have been resized to 32*32 pixels. In .mat file, the struct of data set is shown in Figure 1.2.1, 'Name' is the filename of the corresponding image. One example of the struct of bounding box (bbox) is shown as figure 1.2.1, 'left' and 'top' are the left top point of the bounding box and 'height' and 'width' are the height and width of the bounding box. 'Label' contains numbers in colour images. In label, '1' to '9' means digit '1' to digit '9' in house-number images, '10' represent digit '0'. An example for label structs is shown in Figure 1.2.2. 20% of data is used in SVHN, 13068 digits for training, 33402 digits for training and 202353 digits in the extra dataset.



height	30
left	43
top	7
width	19
label	5

label
2
1
10

Figure 1.2.2

2.Method

2.1 Workflow

Overall, firstly, we implement MSER to find the bounding boxes of different images. Then, we use a convolution neural network as a detector to distinguish which boxes include street number. After that, the Non-Maximum-Suppression (NMS) method has been used to eliminate boxes with large repeat area to improve accuracy and reduce redundancy. In the end, a CNN recognizer has been used to distinguish which numbers are in the bounding box and in which order.

With the help of all these methods, we can obtain the number from a real-world image which includes street plates.

2.2MSER

MSER stands for maximally stable extremal regions, which is used as a method of blob detection in images [4]. This technique is able to find correspondences between image elements from two images with different viewpoints.

In our task, this method has been used to find all the potential bounding boxes for a certain street image. After implementing MSER, we are able to obtain a great number of boxes which include different objects. This can be fulfilling by a built-in function in python. However, there are not only the boxes with street number plate, but also some boxes with maybe trees or houses, which are helpless or even harmful to our number recognition (a figure has shown as below). Therefore, we have to find a method to distinguish which boxes include street numbers.

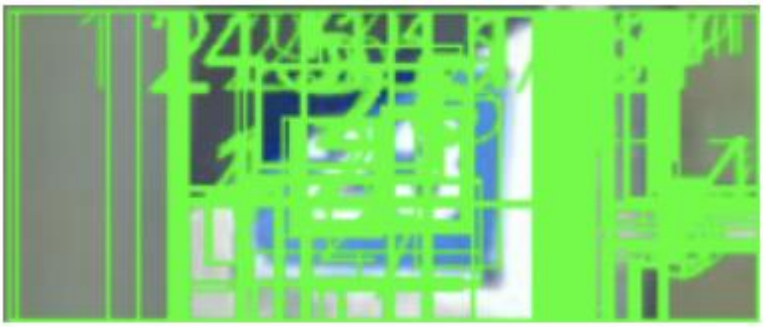


Figure 2.2.1 too many bounding boxes in an image

2.3 CNN Detector and NMS

In order to fulfill the goal, we mentioned at the end of the 2.2 part. A convolution neural network has been built as a detector to distinguish whether a certain box includes street number plate. Our CNN has multiple convolutional blocks, each block has 2 convolutional layers and 1 pooling layers. As the image goes through the CNN, its features are extracted by convolutional blocks and are used to make predictions. The activation function we used is ReLU.

What our model really does is classifying all the boxes into two categories, which are street number plate area and useless area. The model we built here have two convolutions. However, to accomplish this task, we have to get labels for some images to train our CNN model.

Here, we generate the dataset for the detector by comparing our bounding boxes found by MSER with the bounding boxes in the original full image. Note that the cropped images in the dataset are the images with the ground true bounding box which includes street numbers.

For a certain image, we compare the regions found by our MSER method (which we are likely to call MSER regions) and the corresponding image with ground true box provided in our original dataset. If the area overlap rate between a certain bounding box in our MSER image and the correct bounding box is more than a pre-defined threshold (we use 50% here), we are likely to treat this certain box as a street number plate area, and attach the label ‘true’ to it.

After implementing this method, we are able to obtain all the potential bounding boxes area in MSER images and their corresponding labels. We use these data to train our CNN detector. After training, if we input a brand-new image with MSER preprocessing (note that this image is not included in our dataset) and send the regions to the detector, the output of our detector would be the true-false labels for all the bounded areas. In the end, we only take the areas with the label 'true' and abandon the 'false' boxes.

2.4 NMS

After implement and testing our MSER method and CNN detector, we obtain all the boxes which are likely to include street plate. However, some of these bounding boxes are almost the same, which leads to the redundancy problem when recognizing numbers. In order to eliminate repeated boxes, the NMS method has been implemented.

The NMS stands for non-maximum-suppression, which is a method to eliminate redundant candidate boxes. For a certain image, we find the bounding box with largest area and compare all the other boxes in the image. If the IOU (Intersection Over Union between a certain box and the largest box is more than a pre-defined threshold (we use 50% here), we are likely to eliminate the certain box. After comparing all the other box with the largest box, we compare the second largest box with all the other boxes exclude the largest one. Repeat this process after all the boxes has been compared with and all the repeated bounding boxes has been eliminated. Then, we obtained the unique bounding box which includes the street number with no redundancy.

2.5 CNN Recognizer

In this part, a CNN model has been built to classify the bounding boxes into two parts according to whether it includes street number. Our CNN recognizer has the same inner structure as the detector, but the output is the actual number of the image, 1 - 10. Here, we use this CNN model to find all the numbers in the bounding box and recognize what specific numbers there are.

We use all cropped images and the corresponding street number provided by the original data set to train our CNN Recognizer. After our model has been trained well, if we input an image with some bounding boxes, our CNN recognizer is able to find out what numbers are in these boxes. Finally, we can take a picture from the real world, use all the method and model we mentioned here and extract the street number from this image successfully. The figure shows a sample result.

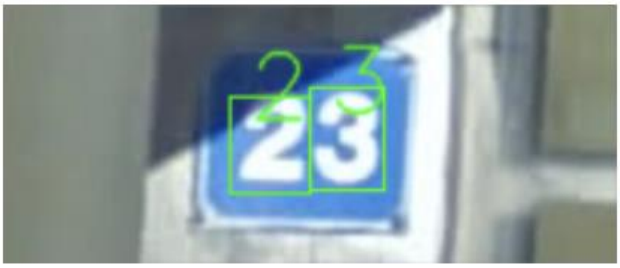


Figure 2.5.1 A sample result of street number recognition

3. Result and Discussion

3.1 Overall

The value of lost express the performance of a certain model after each iteration of optimization, hence it is an important criterion for a model. And the accuracy of a model is determined when the model has finished learning process. It is the calculated percentage of misclassification which is the comparison result of the testing result of the test samples and the true targets. For example, the number of test samples is 100, and model classifies 90 of them correctly, hence the accuracy is 0.9.

Ideally, validation (test) loss starts decreasing when validation (test) accuracy starts increasing which means the model build is learning and everything fine.

3.2 CNN detector

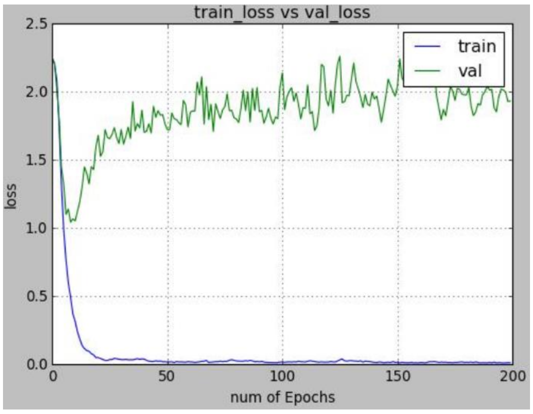


Figure 3.2.1 train_loss and val_loss

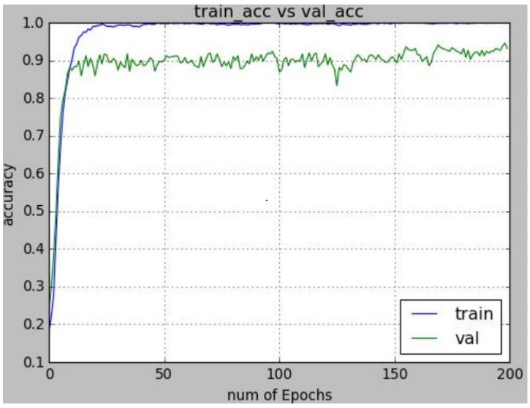


Figure 3.2.2 train_ accuracy and val_accuracy

As shown in the figure 3.2.1 and 3.2.2, In the first few epochs, both of train and validation loss has dropped rapidly, and both of train and validation accuracy has risen sharply. This shows that the model is working well at this point. After that, the validation loss starts to rise gradually, and then fluctuates in a small range after reaching a steady value which is around 2.0. Meanwhile, validation accuracy remained steady at around 0.9 and fluctuates in a small range. The reason of the validation loss keeps increasing while the validation accuracy keeps increasing maybe is because of the calculation of loss is using the actual predicted probabilities and the calculation of the accuracy is using the one hot vector (0,1). In conclusion, the training result is acceptable**.**

3.3 CNN recognizer

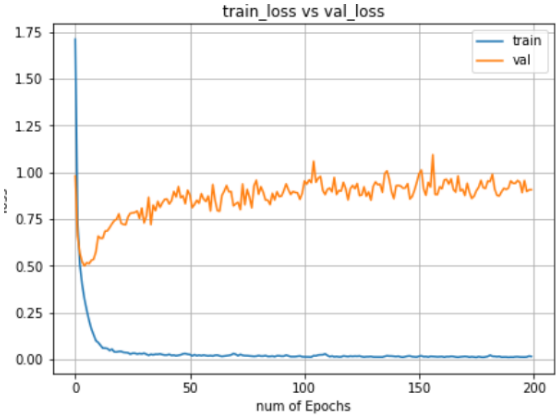


Figure 3.3.1 train loss and val loss

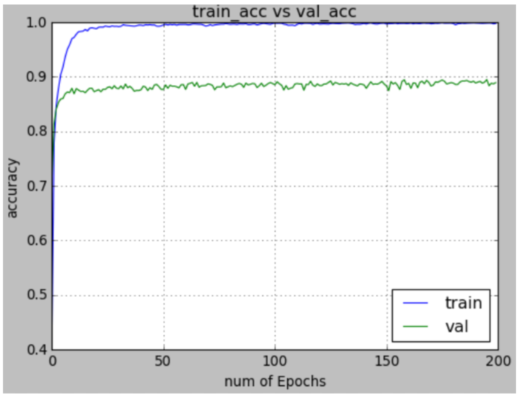


Figure 3.3.2 train accuracy and val accuracy

As shown in the figure 3.3.1 and 3.3.2, In the first few epochs, both of train and validation loss has dropped rapidly, and both of train and validation accuracy has risen sharply. This shows that the model is working well at this point. After that, the validation loss starts to rise gradually, and then fluctuates in a small range after reaching a steady value which is around 0.8. Meanwhile, validation accuracy remained steady at around 0.88 and fluctuates in a small range. I think the reason for the increase in validation loss might have something which is related to the calculation function of the loss. For a good CNN model, the gap between train loss and validation loss should be very low. However, there's some overfitting of the model in general. Hence, it is normal that the train loss is less than validation loss (But the difference between the two cannot be too large). In conclusion, the training result is acceptable.

3.4 the digital recognition process result

By deploying the two models (which are described above), we can get the results shown in figure 3.4.1 with the identified number and its related bounding box.

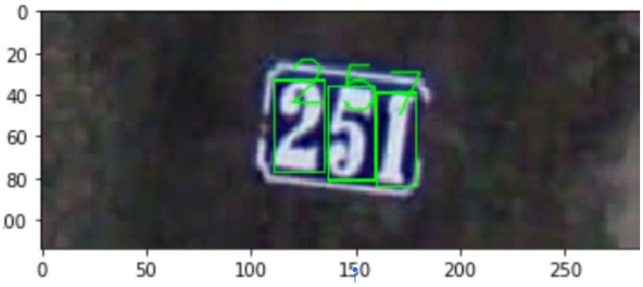


Figure 3.4.1 result

4. Conclusion

In this project, we successfully detect and recognize the street and house number from images and videos. To detect numbers in images, MSER method and a CNN detector are used to find the bounding boxes in images and select the images including street number plates. To recognize the exact numbers, NMS method and a CNN recognizer can eliminate redundant overlapped boxes and obtain specific numbers. The result shows that both detector and recognizer models perform accurately in extracting digits.

According to the experiments, the results show that both models are accurate. The accuracy for the detector is 90%, and the accuracy for the recognizer is 88%.Because we combine the traditional computer version methods such as MSER and NMS, with deep learning model CNN, the images in testing dataset have been preprocessed to obtain clear and large digit area, so the models can predict these images more efficiently.

However, there is still something to improve. The recognizer would consider irrelevant object as a number by mistake, and the accuracy of detector decline when image is rotated. For the future work, the training data set need to include more kinds of images, such as images with rotated and skewed digits. Also, some negative training can be processed to ensure that the detector would distinguish numbers from other objects.

Reference:

[1] Krizhevsky, Alex & Sutskever, Ilya & E. Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[2]G. Huang, Z. Liu, L. v. d. Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2261-2269.

[3]Ming Liang and Xiaolin Hu, "Recurrent convolutional neural network for object recognition," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3367-3375.

[4]H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk and B. Girod, "Robust text detection in natural images with edge-enhanced Maximally Stable Extremal Regions," 2011 18th IEEE International Conference on Image Processing, Brussels, 2011, pp. 2609-2612.

[5]Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, Andrew Y. Ng Reading Digits in Natural Images with Unsupervised Feature Learning NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011.

Peer review:

Juntao Li,	u6342214	20%
Yizhi Zhao,	u6719761	20%
Yuyang Wang,	u6342479	20%
Ben He,	u6321576	20%
Yiman Huang,	u6214187	20%

Did all work together