A Novel Algorithm for Damaged Barcode Recognition Based on Deep Learning

Huijuan Liang, Song Chai, Chuanwu Zhang, Qirong Li, Jiayi Li and Haizhen Wang

ABSTRACT

EAN/UPC barcode is one of the most used barcode system in commodity production, logistics and warehouse operation. However it is unavoidable that the barcode will be damaged during the process of commodity transportation and sale. In this paper, a barcode recognition algorithm based on deep learning is proposed for the recognition of damaged barcode. In our proposal, a convolutional neural network is designed for barcode recognition. The CNN based on deep learning is basically constructed of six convolution layers and three full connected layers. A hundred thousand barcode images with simulated degradation are generated as dataset to train the model, and a custom loss function is utilized to boost the recognition performance. The experiment result shows that recognition rate is up to 99.43%.

KEYWORDS

Barcode Recognition, CNN, Deep Learning.

INTRODUCTION

Barcode system is widely used in product management, commodity transportation and information tracing. EAN/UPC is one of the most used barcode system for commodity, which can be found on every single product[1]. However, it is inevitable that barcode will be damaged during commodity transportation. In this sense, effective monitoring and recognition for these incomplete barcode images becomes essential. Traditional methods of barcode recognition were using linear filtering or median filtering algorithm, which was not stable and has low recognition rate of damage barcode.

In this paper, a barcode recognition algorithm based on deep learning is proposed. The CNN is constructed of six convolution layers and three full connected layers, dropout and max pooling are also added to enhance the capacity of the model, and custom loss function is utilized to boost the barcode recognition performance.

In our proposal, a hundred thousand barcode images with simulated degradation are generated as the dataset of deep learning, 90,000 among them are to train the model, and the rest 10,000 are for model testing, the recognition success rate is up to 99.43%.

The rest of this paper is organized as follow: Section 2 summarizes some of the re-
lated work, our algorithm is proposed in Section 3, experiment and results are given in Section 4 and Section 5 concludes the paper.

RELATED WORK

Ventsov, N. N. and L. A. Podkolzina proposed a parallel video stream processing method for the linear barcode recognition. Improvements were made to solve the problem of a more efficient identifying of the objects in video stream[2].

Hansen, Daniel Kold and et al. described how to adapt the deep learning-based detector of YOLO for the purpose of detecting barcodes in a fast and reliable way. The detector achieved state-of-the-art results on the benchmark dataset of Muenster BarcodeDB with a detection rate of 0.991[3].

Fernandez, Wendy P., Yang Xian, and Yingli Tian proposed a barcode detection and recognition method for a shopping assistant. The proposed method was capable of extracting the essential product information from the detected barcode region[4].

Kim, Young Jung, and Jong Yun Lee proposed a novel segmentation and normalization method in PDF417 with the aims of improving its recognition rate and precision, the method showed a stable performance over existing PDF417 barcode detection and recognition[5].

Chen J, Pu X, and Guo H, et al. designed a triboelectric based barcode recognition system for practical applications. By using a reference barcode component, the output signal under random swiping motion could be easily recognized[6].

PROPOSED ALGORITHM

In this section, we present the proposed barcode recognition CNN model. Generally training a CNN model for a specific task generally involves two steps: (i) network architecture design and (ii) model learning from training data[7]. For network architecture design, a new CNN architecture is designed and aimed at the barcode recognition task. For model learning, we adopt the custom loss function, and incorporate it with dropout, max pooling and batch normalization for training and improving barcode recognition performance. In the meanwhile, the loss function will be presented as follows.

CNN ARCHITECTURE

In our proposal, the CNN architecture is mainly designed with two types of layers, shown in Figure 1 with two different colors.

To capture underlying features of barcode images, the CNN model sequentially works as Figure 1. shown. (i) Conv+ReLU: 64 filters of size $3 \times 3 \times c$ are used to generate 32 feature maps, and rectified linear units (ReLU) are then utilized for nonlinear[8]. Here $c$ represents the number of images channels, and in our implementation $c=1$ is chosen for gray image. (ii) FC: The layer with size of 1024 is connected to the convolution output, and batch normalization is incorporated to speed up training as well as boosting the barcode recognition performance. This layer is used
to reconstruct the output. Note that dropout and max pooling are also added to enhance the capacity of the model.

The input of the CNN model is barcode images with size of 339*169, and its output should be the correct recognition code letter that provides the information of barcode. Note that the 13 digits in EAN-13 code are not mutually exclusive, which can be regarded as Multiclass Classification task[9]. On the other hand, the probability of any number in 0~9 about each number among the 13 is mutually exclusive, which can be regarded as Multilabel Classification task. In this way, the input barcode image can transfer into the output format with a 13*10 matrix, which is named H(X). Directly flat it into a 130*1 matrix for facilitating calculation, in this matrix every ten matches a digit of the 13, and each one in the ten matches the possibility of any number in 0~9 about each digit. Therefore, the output contains 130 neurons which match our model. As the barcode recognition is multi-task classification[10], the softmax function is adopted to calculate the row output of full connected layer after the 130*1 matrix reconstructing into a 13*10, where the i-th row represents the softmax output of the i-th number of the 13 code digits.

Figure 1. The CNN model.

CUSTOM LOSS FUNCTION

As mentioned above, barcode recognition task is not a multilabel classification nor a multiclass classification task. Therefore a custom loss function \( L(\theta) \) is required for training the model. The following is about the custom loss function using in the algorithm.

\[
L_i(\theta) = P(y=y^i|x=x^i;\theta) = P(y_0 = y_0^i)P(y_1 = y_1^i)\cdots P(y_{12} = y_{12}^i)
\]  

(1)

Here \( y_0 \) donates the first number among those 13 digits, \( y_1 \) donates the second one, and so on. The superscript (i) of x and y represents the i-th sample in the training set.

After applying log to the both side of (1), the cost of the i-th sample is:

\[
I_i(\theta) = \log L_i(\theta) = \log \left[ P(y_0 = y_0^i) + P(y_1 = y_1^i) + \cdots + P(y_{12} = y_{12}^i) \right]
\]  

(2)

Considering total m samples in the dataset, the loss function is:

\[
L(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[ \log \left[ P(y_0 = y_0^i) + P(y_1 = y_1^i) + \cdots + P(y_{12} = y_{12}^i) \right] \right]
\]  

(3)
In the above loss function, $P(y_d = y_d^{(i)})$ is the cross entropy loss of the d-th digit in EAN-13. Hence, equation (3) can be rewritten as:

$$L(\theta) = -\frac{1}{m} \sum_{i=1}^{m} \left[ \sum_{k=0}^{9} \log h(y_d^{(i)} = k) \right]$$

(4)

MODEL TRAINING

A complete barcode includes black-and-white stripe and the number below. According to this, a ready-made Python package is imported to generate a large amount of barcode images as the dataset of our model. Then one hundred thousand images are generated and saved as raw dataset. Next four kinds of noise consisting of ‘gauss’, ‘posion’, ‘s&p’ and ‘speckle’ are defined and added to emulate the blurred barcode images, here the ‘gauss’ is Guassian-distributed addictive noise, ‘posion’ equals Possion-distributed noise generated from data, ‘s&p’ is noise replacing random pixels with 0 or 1, ‘speckle’ donates multiplicative noise using ‘out = image + n*image’ where n is uniform noise with specified mean and variance.

Similarly the random black and white spots are defined to emulate the damaged barcode, what need to be mentioned is the sizes, colors and locations of these spots are random and the spot number is between 3 and 12. Then deterioration treatment begins for 80% among the raw dataset. The four kinds of noise of any two are overlapped to 30,000 barcode images as blurred barcode. Some black and white spots are randomly generated on another 30,000 barcode images to be the damaged barcode. Take another 20,000 barcode images overlapped with all elements as mentioned earlier as the strong damaged barcode. As of now, the whole dataset is completed, which contains of 20,000 clean barcode, 30,000 blurred barcode, 30,000 damaged barcode and 20,000 strong damaged barcode. Some samples of it are shown in the Figure 2. Then the dataset is randomly split into two groups: 90,000 samples of training set and 10,000 samples of test set.

EXPERIMENT AND RESULTS

Our algorithm is implemented using Python3.6.5 on a PC with dual Intel Xeon E5-2643 + 32GB DDR4 RAM and GTX 1080 Ti graphic card. PyStrich 0.8 is used to generate barcode images as dataset, the data preprocessing is performed using Numpy 1.14.3 and the model is implemented using Tensorflow-GPU 1.10.0.

PARAMETER SETTING

Set the max pooling size to 2*2, and the dropout for 0.2 to faster training and reduce over-fitting. As is mentioned before, the $L(\theta)$ is utilized to calculate the output of the three full connected layers. Generally an activation function should be attached to the final output, there is none being attached in our CNN model for the barcode recognition task is not a multilabel classification or a multiclass classification task.
Therefore a custom loss function $L(\theta)$ is required for training the model. The final 130 output of full connected layer are reconstructed into a $13 \times 10$ matrix, and a softmax function is utilized to calculate every row mapping the 13 code digits on the EAN-13 barcode, where the i-th row donates the softmax output of the i-th number of the 13 code digits.

**RESULTS**

After training the model, the remaining 10,000 barcode images are used to test the recognition rate. The experiment result shows that only 57 of 10,000 barcode images in the test set cannot be recognized, which means the accuracy of this model is up to 99.43%. Some of the unrecognized barcode image is shown in Figure 3. Note that the majority barcode images with seriously damaged are showed, which equals that the lightly blurred or damaged barcode can easily recognized for the model.

![Figure 2. Some sample images of the dataset.](image)

![Figure 3. Sample of experiment result.](image)
CONCLUSION

In this paper, a barcode recognition algorithm based on deep learning is proposed. The algorithm is aimed at those damaged barcode images and solving their recognition problems. In our proposal, a convolutional neural network is designed for barcode recognition. A dataset contains of one hundred thousand barcode images with simulated degradation is generated for model training, and a custom loss function is utilized for the specific barcode task to train the model. With the corporation of max pooling, dropout and batch normalization, the model based on deep learning shows a better performance. While for barcode images with different damage, the recognition success rate can up to 99.43%.

REFERENCES

1. Paunescu, Daniela, Wendelin J. Stark, and Robert N. Grass, "Particles with an identity: tracking and tracing in commodity products," Powder technology 291, 2016, pp. 344-350.
2. Ventsov, N. N., and L. A. Podkolzina, "Studying the effect of paralleling settings on the functioning of a barcode recognition app," 2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). IEEE, 2017, pp. 1-5.
3. Hansen, D. K., Nasrollahi, K., Rasmussen, C. B., and Moeslund, T. B., "Real-time barcode detection and classification using deep Learning," IJCCI. 2017, pp. 321-327.
4. Fernandez, Wendy P., Yang Xian, and Yingli Tian, "Image-based barcode detection and recognition to assist visually impaired persons," 2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER). IEEE, 2017, pp. 1241-1245.
5. Kim, Young Jung, and Jong Yun Lee, "Algorithm of a perspective transform-based PDF417 barcode recognition," Wireless Personal Communications 89.3 (2016): 893-911.
6. Chen, J., Pu, X., Guo, H., Tang, Q., Feng, L., Wang, X., and Hu, C., "A self-powered 2D barcode recognition system based on sliding mode triboelectric nanogenerator for personal identification," Nano Energy 43 (2018): 253-258.
7. Schmidt-Hieber, Johannes, "Nonparametric regression using deep neural networks with ReLU activation function," arXiv preprint arXiv:1708.06633, 2017.
8. Ide, Hidenori, and Takio Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017, pp. 2684-2691.
9. Cheng, D., Gong, Y., Zhou, S., Wang, J., and Zheng, N., "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1335-1344.
10. Wei, Guiwu, "Picture fuzzy cross-entropy for multiple attribute decision making problems," Journal of Business Economics and Management 17.4 (2016): 491-502.