https://doi.org/10.2352/ISSN.2470-1173.2019.8.IMAWM-413

Barcode Detection and Decoding in On-line Fashion Images*

Qingyu Yang[1], *Gautam Golwala*[2], *Sathya Sundaram*[2], *Perry Lee*[2], *Jan Allebach*[1]
[1] **Purdue University, West Lafayette, Indiana 47906, U.S.A**
[2] **Poshmark Inc., Redwood City, CA 94065, U.S.A**

Abstract

A barcode is the representation of data including some information related to goods, offered for sale which frequently appears on manufactured items. Especially in the online fashion market such as Poshmark (a second-hand fashion market), barcodes on the tags of the sale items represent the identified information including producer, manufacturer, etc. The market needs a system to automatically detect and decode barcodes in real time. However, the existing methods have some limitations for detecting 1-D barcodes in various backgrounds including tassels, stripes, and clustered text in fashion images. In this research, our focus is on identifying the barcodes in fashion images and distinguishing the barcode from similar non-barcode image content. It is accomplished by applying a Convolutional Neural Network (CNN) to solve this typical objective detection problem. A comparison of the performance between our algorithm and a previous method will be given in our results. Also, a traditional method based on hand-crafted features will be proposed for comparison. For the decoding part, a package including current common types of decoding schemes is used in our work to decode the detected barcodes. But it fails to decode strongly skewed barcode images. Adding pre-processing to warp the skewed images is used to increase the success of decoding.

1. Introduction

Barcodes always carry essential information for the corresponding manufactured products. With the significance and the variety of applications, detecting and decoding barcodes is broadly useful for online shopping platforms. As a result, these retailers need to process a tremendous number of images containing barcodes. But it would be labor-intensive to label and decode barcode manually. Thus, a system is for detecting and decoding barcodes are widely used. In this work, the general goal to develop a robust automatic system to detect and decode barcodes. Currently, the industries select different types of barcodes for manufactured items such as 1-D and 2-D barcodes. As the 1-D barcodes have longer history and are more commonly used in fashion markets, this research only focuses on the 1-D barcodes. As a 1-D barcode represents specific data by the varieties of widths and spacings of its parallel lines, the detecting algorithms and the decoding schemes are based on these characteristics.

There are a lot of previous works on detecting rotated, obscure, and occluded barcodes with traditional method [1] [2] [3]. In 2015, Faster R-CNN has been proposed by S. Ren, et al. [4]. In 2017, J. Li, et al. implement Faster R-CNN to detect 1-D barcode and get higher accuracy than the traditional methods [5]. Most previous works have been based on datasets that include at least one barcode and clear backgrounds in each image. However, some common patterns used in fashion design such as tassels, brand logos, cells, and stripes are misclassified as barcodes in fashion images as shown in Fig. 1. In other words, the more complicated backgrounds in online fashion images could bring the detection task into a more challenging phase. To face the challenge in barcode detection, the focus will be distinguishing barcodes from the similar non-barcode images. we treat similar non-barcode patterns such as stripes as hard negative samples in this task. Adding the hard negative samples is proposed to implement a CNN model to overcome the previous limitations. For comparison, the development of a traditional method based on specified hand-crafted features is also proposed to solve this task. Then a comparison of algorithm performance among the proposed method, the previous deep learning method, and the traditional method will be shown based on the same testing dataset.

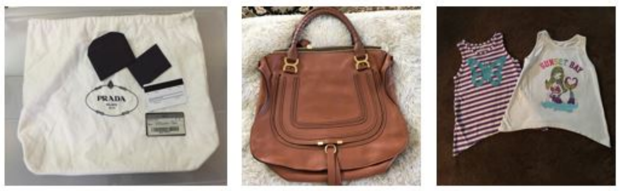


Figure 1: Fashion images with brand logos, tassels, and stripes.

To obtain information from the barcode, the decoding process is important in the development. Based on the scheme of different barcode types, the decoding solution is generated by transferring the number of pixels in parallel lines to a digital number [6] [7]. In addition, there is an online open source Pyzbar in a Python package developed by L. Hudson [8]. This package can decode common types of barcode. Also, lightly skewed and occluded barcode images can also be decoded directly without any pre-processing. However, the existing systems of decoding barcodes still have some limitations with skewed, small, obscure, and rotated barcodes. For example, scanning always fails when the detecting device is far away from barcode; and sometimes decoding strongly skewed barcode images also fails by using the current package directly. A pre-processing method for the strongly skewed images based on computer vision techniques is proposed in this research to solve the previous limitations on decoding barcodes.

*Research partially supported by Poshmark, Inc.. Redwood City, CA 94065

2. Method

The system for detecting and decoding barcodes is constructed as two major parts. The first part is barcode detection. Then a non-rotated enclosing rectangle surrounding the detected region is cropped for the decoding subsystem. Figs. 2 (a) - (d) also display an example of processing an image that includes a barcode on the tag. Our method of applying a deep learning approach will be firstly introduced. Then there will be a description of our development of a traditional method based on hand-crafted features. The last part of the paper discusses barcode decoding with the method of warping the strongly skewed images.

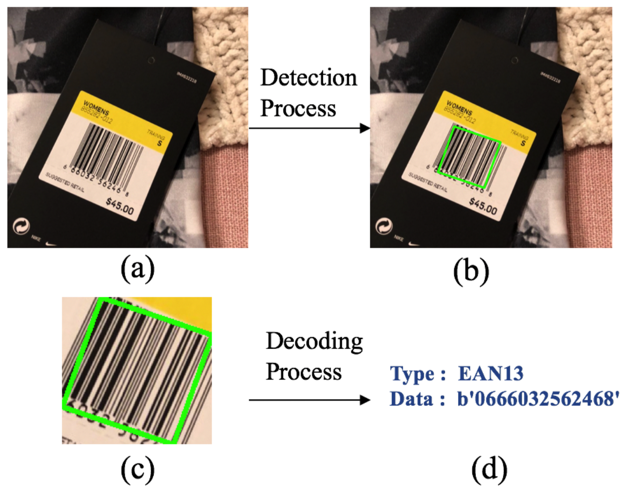


Figure 2: An example of detecting and decoding a fashion image (a) input image generated from Poshmark website; (b) labeling result of barcode detection; (c) barcode image from cropping bounding box; (d) result of barcode decoding.

2.1 Detection with deep learning approach

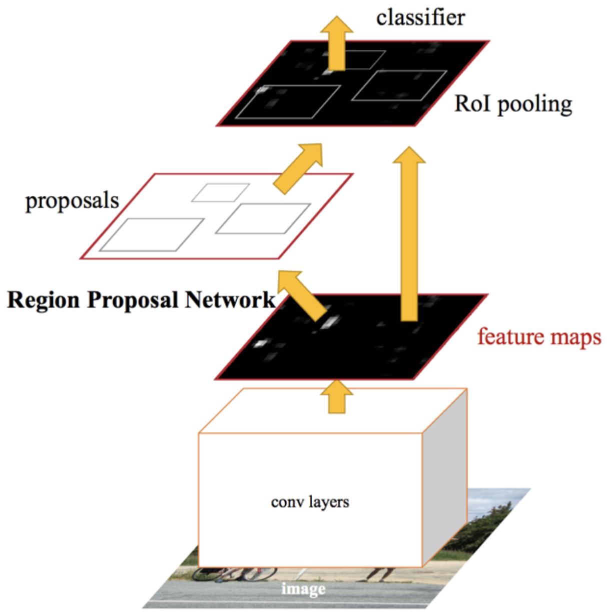


Figure 3: Architecture of Faster R-CNN.

In Faster R-CNN, the RPN is a one-layer network for the region proposal, and the classifier is the network for identifying the barcode. After the convolutional layers, the system maps the input image to a high-dimension space called the feature map. The function of the RPN to select regions containing features of the barcode. For each feature map, the RPN will propose 9 different sliding-windows with different sizes, as shown in Fig. 4. Those selected regions in the feature map will return to the input image. Therefore, thousands of bounding boxes will be generated in the input image. Also, the classifier will assign a score for each region. The score represents the probability of the region corresponding to a barcode. Fig. 4 shows an example of the generated bounding boxes from RPN. In this example, many regions around the face will also be assigned a very high score. Then non-maximum suppression (NMS) is used to reduce the generated regions around the object to a single region with a bounding box. The algorithm is shown in Fig. 5. If the value of Intersection over Union (IoU) is larger than a threshold, the regions with the smaller score will be removed to make the training process faster. In our design, we set the threshold to be 0.7. Also, the classifier in Faster R-CNN consists of fully-connected layers. In our case, it regresses all the generated features in the region to a single number.

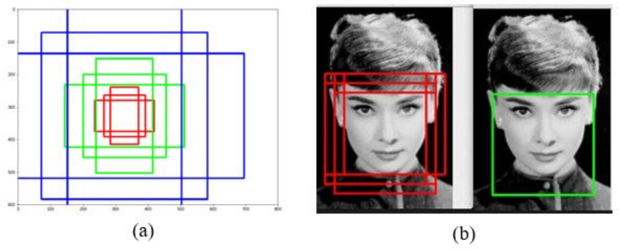


Figure 4: Sliding windows with different sizes.

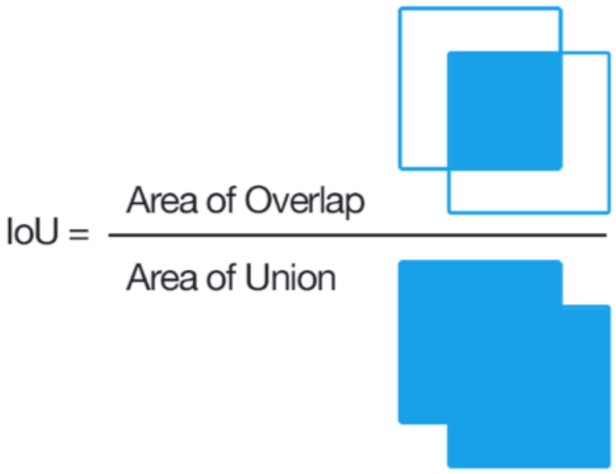


Figure 5: Function of Intersection of Union.

The single class, which is "barcode" is used in the first experiment. It shows that the CNN model is robust and achieves a good result. However, the first experiment still has some failure cases. We find that the networks sometimes misclassify the fashion items that contain stripe patterns as barcodes. The reason is that the stripe patterns and the barcodes share many similar visual features. Thus, the stripe patterns are hard negative samples in this task to help the networks to distinguish between the stripe patterns and the barcodes. In our design, we modifying the networks to recognize both barcodes and stripe patterns. Based on this design, the feature space also contain more features to help distinguish between barcodes and stripe patterns.

To implement the above design, the training dataset has 700 images, which contain 500 barcode images and 200 images with stripes. To better match the goal, which is detecting barcodes in fashion images, the dataset is generated by taking pictures containing barcodes with varying backgrounds at fashion stores. The backgrounds include kinds of elements common to images containing fashion items, such as clothes, floors, packages, etc. The training dataset also includes some small, rotated and occluded barcodes with different camera angles and varying illumination. Also, the ground truth of the 700 training images is created by manual labeling. An online open-source toolbox [9] to record bounding boxes is used to label the barcode regions and the stripes with the two different classes for each training image.

In addition, the hyper-parameters need to be specified before training the Faster R-CNN. The applicable hyper-parameters would help the model to achieve better performance and reduce the time cost in the training process. To help the network to distinguish between barcodes and stripe patterns, two classes with "barcode" and "stripes" are used in the training process. Since the convolutional layers use the pre-trained model [10], a small training rate, 0.0002, is used in the training [11]. Also, the total number of images in the training dataset is 700, which will not be an issue for computation time. Thus, the images are fed to the training system one-by-one. The number of global steps in the training process is around 13000 to guarantee that the training process has converged. In the classification, 93% is set to be the threshold of confidence value for higher accuracy. It means that only the region assigned with a higher confidence score than the threshold can be selected to be the barcode region.

For comparison, the previous deep learning method with only one class ("barcode") has also been implemented. The previous method only requires the network to identify the barcodes. Thus, the training dataset for this implementation only has 500 images with the barcodes. The generation of ground truth is done in the same way as in the above description. The hyper-parameters except for the number of classes are also kept the same as mentioned above. In other words, this research repeats the experiment mentioned above without the "stripe" ground truth images.

2.2 Detection with the traditional method based on hand-crafted features

To compare the performance with the deep learning approach, the traditional methods are also implemented and developed to achieve the goal. Many previous works on barcode detection with the traditional method use combinations of detectors and morphological operations [12]. Based on the previous work, this research also creates a system to detect barcodes with a combination of edge detection, morphological transformation, and post-processing of hand-crafted features. In this system, the first part uses edge detection and morphological transformation to find the largest foreground cluster called C_{max} in a processed binary image. In addition, the pattern in the input image that corresponds to C_{max} is called C_i. The second part is post-processing to identify whether C_i is a barcode or not.

Fig. 6 shows an example of detecting a barcode in a fashion image with the traditional method. As the barcode consists of many parallel lines, the small edges between the lines can be detected by edge detection. Firstly, the gradient image is used to detect the edges of the input image. Because the gradient image can display the change in intensities, the small edges of barcodes could be easy to detect. It is computed by applying the Sobel operator [13] with a 9×9 kernel to the grayscale image. Also, an averaging filter and a threshold will be used to get the binary image, which can split the foreground and background. The binary image is shown in Fig. 6 (b). It can be seen that the region corresponding to the barcode has some small gaps. Thus, morphological operations, erosion and dilation, are used to fill the gaps. As shown in Fig. 6 (c), the largest foreground cluster in the binary image will have the largest probability to correspond to a barcode in the input image.

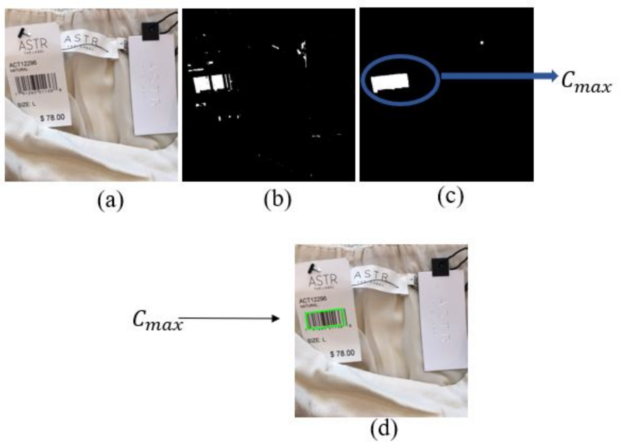


Figure 6: An example of the detection process with the traditional method.

The identification of the largest foreground cluster C_{max} in the input image will be realized by our development: adding post-processing. The post-processing is constructed by setting 4 thresholds ($T_1 - T_4$). According to the comparison of the hand-crafted features ($feature_1 - feature_4$) and the tuned thresholds, C_i will be predicted as a barcode or not. Setting all of the features and the thresholds is based on the characteristics of the typical 1-D barcode. The following describes the 4 features in the post-processing.

- Area of Barcode T_1
  From the 200 testing results, some failure cases in the images without a barcode have a small bounding box. Thus, the feature which represents the dimension of the largest foreground cluster is used to check if C_{max} corresponds to a visible barcode. As shown in Equation 1, if the number of pixels in C_{max} is larger than T_1, the C_i will pass the first check.

$$feature_1 = number\ of\ pixels\ in\ C_{max} \tag{1}$$

- Shape of Barcode T_2
  It is visible that the contours of typical barcodes are similar to a rectangle. With the general shape of the barcode, we select the second threshold based on the comparison between the shape of C_{max} and a rectangle. $feature_2$ is the ratio of the number of pixels in the bounding rectangular area and the number of pixels in C_{max}, as shown in Equation 2.

$$feature_2 = \frac{number\ of\ pixels\ in\ C_{max}}{number\ of\ pixels\ in\ bounding\ rectangle} \tag{2}$$

- Corners in Barcode T_3
  A barcode is formed by a black-and-white pattern and includes dozens of corners. Thus, detecting corners also helps us to identify if C_i is a barcode. We check this by applying the Harris corner detector [14]. The ratio of the number of detected corners and the number of pixels in C_i is set to be checked as defined by Equation 3. If C_i includes a sufficient number of corners that $feature_3$ larger than T_3, it will pass the third check.

$$feature_3 = \frac{number\ of\ corners\ in\ C_{max}}{number\ of\ pixels\ in\ C_{max}} \tag{3}$$

- Color Information in Barcode T_4
  In the real world, barcode shown in color image is always a 2-D color image which has three channel R, G, and B. But it always looks like a binary barcode. From our analysis, the real barcode region has a smaller difference between R, G, and B values than the colorful stripes associated with fashion garments. Thus, we can calculate the average difference between the R, G, and B channels as shown in Equation 4 to approximate the colorfulness of C_i. N represents the number of pixels in C_i

$$feature_4 = \frac{\sum_{j=1}^{N}((R-G)+(R-B)+(G-B))}{N} \tag{4}$$

2.3 Barcode decoding

To decode the barcode, the Pyzbar package is used in this work to identify the barcode types and obtain the information. The Pyzbar package in Python is used to decode common types of barcode. This package is set up with different decoding schemes, barcode correlation, and pre-processing techniques. Thus, even though the barcode is slightly skewed, it can also be correctly detected by the Pyzbar package. Also, it can identify the type of input barcode and decode the information according to the schemes of the different types.

For some images including a strongly skewed barcode, Pzybar cannot decode the barcode directly. The current software on our mobile phone also cannot decode it. We use perspective transformation [15] based on the four corners on the barcode to warp it. Fig. 7 shows the process.

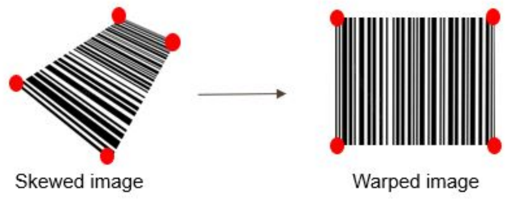


Figure 7: Process of warping the strongly skewed image.

The perspective transformation projects an original image to a new projective plane. It is realized by matrix calculation (Equation 5) of the coordinates. Here, $[u,v]^T$ represents the coordinates in the original image, and $[x,y]^T$ represents the coordinates in the projective plane. The matrix H represents the general projective transform. The h_{33} equals 1. The matrix $\begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix}$ is the matrix of a linear transformation including scaling, shearing, and rotation. Due to the linear transformation, $[h_{31}\ h_{32}]$ is set to zero. $[h_{13}\ h_{23}]^T$ represents the translation. The coordinates of the labeled corners are used to generate the matrix H.

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \times \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad x = \frac{x'}{w'} \quad y = \frac{y'}{w'} \tag{5}$$

3. Experiment

3.1 Barcode detection results

One thousand images were randomly downloaded from the Poshmark website for testing the performance. So the testing dataset includes online fashion images with barcodes or without any barcode. The barcodes in the images are expected to be labeled by green bounding boxes. As the networks also need to classify some stripes, many stripe patterns are labeled by yellow bounding boxes. However, only green bounding boxes, which are labeling barcodes, will be analyzed in the testing algorithm performance. For this objective detecting problem, the typical values of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) are used in the analysis. For example, the TP represents the successful case that the input image includes at least one barcode and the system bounds the barcode in the image correctly. Fig. 8 shows some examples of successful cases; and Fig. 9 shows some examples of failure cases. The most cases of false negatives are due to the small areas of barcodes such as the bottom images in Fig. 9. The red circles in the images in the bottom row of Fig. 9 are the ground truths. The dimensions of the barcodes in these images are around 8×57 and 10×40, respectively. But the dimensions of the labeling boxes are required to be larger than 32×32. This results in a limitation for detecting very small barcodes.

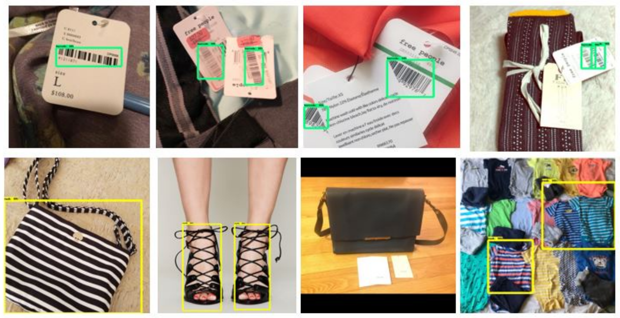


Figure 8: Examples of the true positives and the true negatives in barcode detection with the deep learning method.

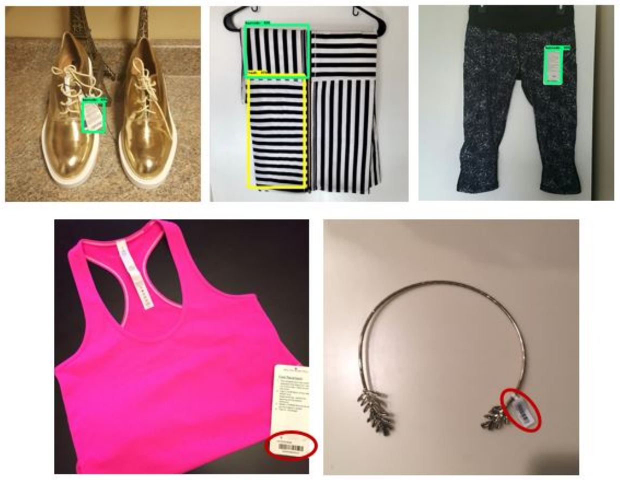


Figure 9: Examples of the false positives and the false negatives in barcode detection with the deep learning method.

For comparison, all of the mentioned three methods, including the deep learning approach with and without "stripes" class, and the traditional method are tested on the same 1000-image dataset. The accuracy *Acc* is calculated by Equation 7. Also, the sensitivity [16] is a value that represents the ability of the system to correctly detect the barcodes. A comparison of the three methods is given in Table 1. Here, N_{test} is 1000, which is the number of testing images.

$$Acc = \frac{N_{TP} + N_{TN}}{N_{test}} \tag{6}$$

$$Sensitivity = \frac{N_{TP}}{N_{TP} + N_{FN}} \tag{7}$$

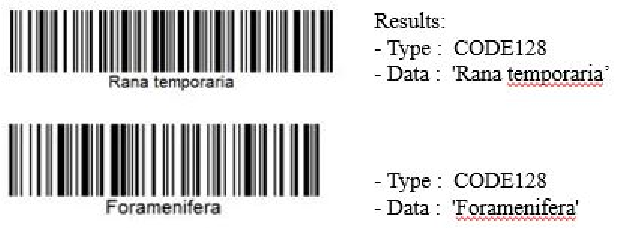
It is evident that adding the new class "stripes" helps to solve the false positives and increase the accuracy and sensitivity. Our development on the deep learning achieves the highest accuracy. The traditional method takes the shortest running time, but it has lower accuracy and more false positives than the deep learning approaches.

Table 1: Accuracy comparison of three methods for barcode in detection

Method	Acc	Run Time	N_{FP}/N_{test}	Sensitivity
Developed deep learning	96.7%	169.23s	12/1000	87.0%
Previous deep learning	92.4%	143.86s	49/1000	83.3%
Traditional method	82.8%	44.52s	83/1000	45.1%

3.2 Barcode decoding results

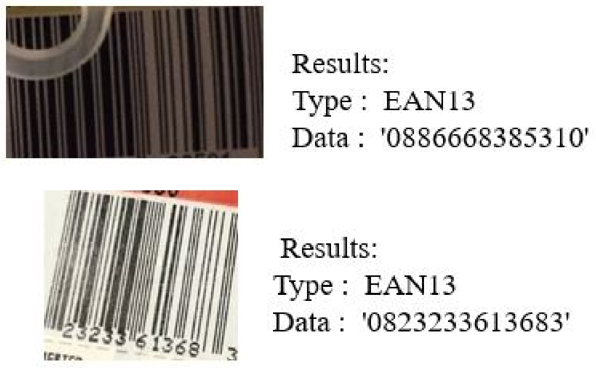
The Pyzbar package in Python has a good performance on decoding the barcodes. Fig. 10 shows the results from the decoding by the Pyzbar package directly. Also, Fig. 11 displays the decoding results from the detected region after the barcode detection system. In this result, the occluded and slightly skewed barcode images can also be detected directly.



```
Results:
- Type :  CODE128
- Data :  'Rana temporaria'

- Type :  CODE128
- Data :  'Foramenifera'
```

Figure 10: Decoding results from the Pyzbar package with the different types of barcodes.



```
Results:
Type :  EAN13
Data :  '0886668385310'

Results:
Type :  EAN13
Data :  '0823233613683'
```

Figure 11: Decoding results from the Pyzbar package with the detected barcode region.

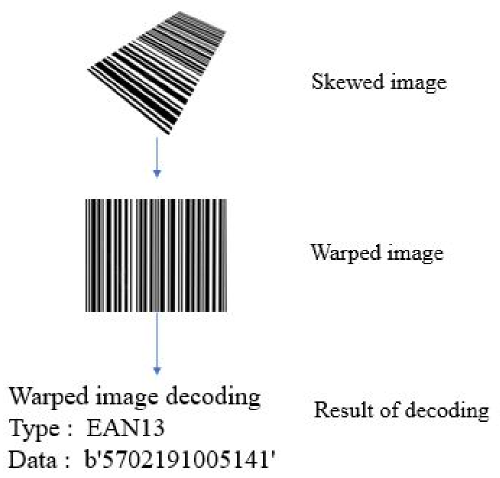


Figure 12: Decoding results from the Pyzbar package after warping a barcode image.

For the strongly skewed barcode images, the Pzybar package cannot decode it directly. After applying the four-point perspective transformation, Pzybar can successfully decode the warped image as shown in Fig. 12.

4. Conclusion

The developed system is desirable to identify images of fashion items that contain barcodes and to decode those barcodes. Also, the challenge that fashion items frequently contain stripes that can be confused with barcodes has been solved in the development. We explicitly consider stripes as detection class, along with barcodes. In this work, the three approaches to barcode detection have been implemented. From testing, the comparison is observed for those three methods in run time and accuracy. The proposed method with the explicit class for stripes with the deep learning approach achieves the highest accuracy of 96.7%. To decode the significantly skewed barcodes, the Pyzbar package has been applied after warping the barcode image.

References

[1] H.-T. Li, J. Li, H. Hwang, X. Jiang, and J. Cheung, "Barcode detection based on morphological operations," Feb. 5 2013, US Patent 8,366,004.

[2] M. Katona and L. G. Nyúl, "A novel method for accurate and efficient barcode detection with morphological operations," in *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*. IEEE, 2012, pp. 307–314.

[3] P. Bodnár and L. G. Nyúl, "Improving barcode detection with combination of simple detectors," in *2012 Eighth International Conference on Signal Image Technology and Internet Based Systems*. IEEE, 2012, pp. 300–306.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[5] J. Li, Q. Zhao, X. Tan, Z. Luo, and Z. Tang, "Using deep convnet for robust 1d barcode detection," in *International Conference on Intelligent and Interactive Systems and Applications*. Springer, 2017, pp. 261–267.

[6] H. I. Hahn and J. K. Joung, "Implementation of algorithm to decode two-dimensional barcode pdf-417," in *6th International Conference on Signal Processing, 2002.*, vol. 2. IEEE, 2002, pp. 1791–1794.

[7] R. Muniz, L. Junco, and A. Otero, "A robust software barcode reader using the Hough transform," in *Proceedings 1999 International Conference on Information Intelligence and Systems (Cat. No. PR00446)*. IEEE, 1999, pp. 313–319.

[8] L. Hudson, "Pyzbar: Read one-dimensional barcodes and QR codes from Python 2 and 3." [Online]. Available: https://pypi.org/project/pyzbar/

[9] "How to train a TensorFlow Object Detection Classifier for multiple object detection on Windows." [Online]. Available: https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10

[10] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "CNN: Single-label to multi-label," *arXiv preprint arXiv:1406.5726*, 2014.

[11] G. Baldassarre, "A modular neural-network model of the basal ganglia's role in learning and selecting motor behaviours," *Cognitive Systems Research*, vol. 3, no. 1, pp. 5–13, 2002.

[12] A. Plaza, P. Martínez, R. Pérez, and J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 9, pp. 2025–2041, 2002.

[13] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, "Design of an image edge detection filter using the Sobel operator," *IEEE Journal of Solid-State Circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[14] C. G. Harris, M. Stephens *et al.*, "A combined corner and edge detector," in *Alvey Vision Conference*, vol. 15, no. 50. Citeseer, 1988, pp. 10–5244.

[15] J. Mezirow, "Perspective transformation," *Adult Education*, vol. 28, no. 2, pp. 100–110, 1978.

[16] D. G. Altman and J. M. Bland, "Diagnostic tests. 1: Sensitivity and specificity." *BMJ: British Medical Journal*, vol. 308, no. 6943, p. 1552, 1994.

Author Biography

Qingyu Yang received her B.S.(2017) in Electrical Engineering in Purdue University and currently is a second-year master student from Purdue ECE. Her research focuses on image quality analysis, computer vision, objective detection, and deep learning applications.

Jan P. Allebach is Hewlett-Packard Distinguished Professor of Electrical and Computer Engineering at Purdue University. Allebach was named Electronic Imaging Scientist of the Year by IST and SPIE, and was named Honorary Member of IST, the highest award that IST bestows. He has received the IEEE Daniel E. Noble Award, the IST/OSA Edwin Land Medal, is a Fellow of the National Academy of Inventors, and is a member of the National Academy of Engineering.

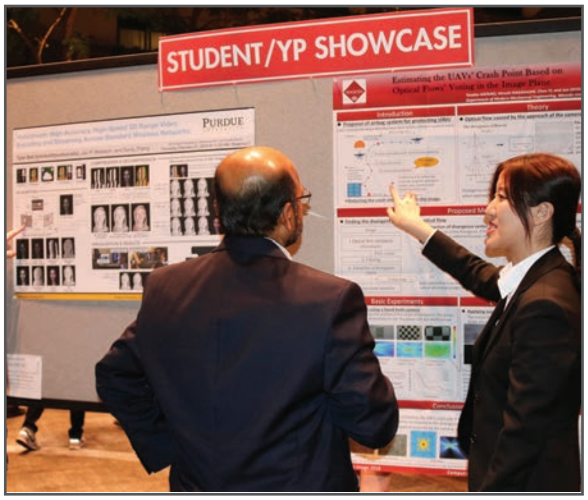
JOIN US AT THE NEXT EI!

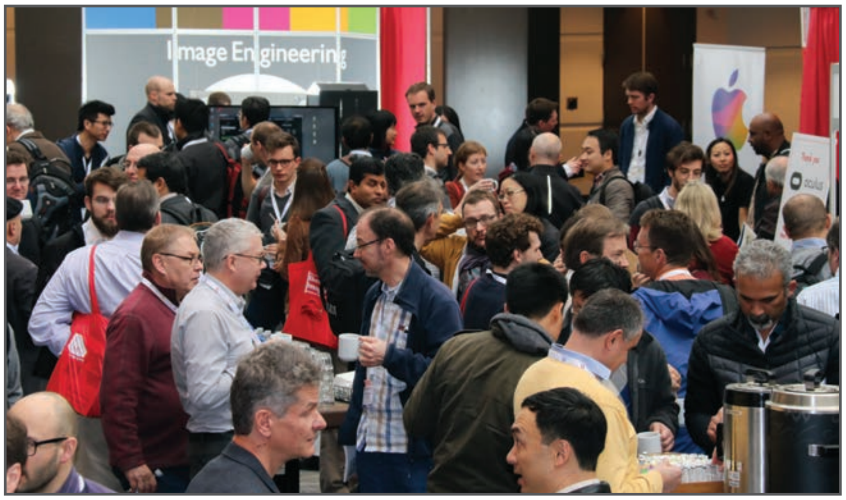
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





AR is a fundamental and transformative interface and platform.
20+ years from my initial working demo to today's AR displays
Optimistic that it won't take another 20 to reach ubiquitous consumer AR!
My first working AR prototype
Today's platforms
Ubiquitous consumer AR

• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org