15-08

15th IAPR International Conference on Machine Vision Applications (MVA)
Nagoya University, Nagoya, Japan, May 8-12, 2017.

Banknote Portrait Detection Using Convolutional Neural Network

Ryutaro Kitagawa*, Yoshihiko Mochizuki*, Satoshi Iizuka*, Edgar Simo-Serra*,
Hiroshi Matsuki†, Naotake Natori†, Hiroshi Ishikawa*
*Waseda University, Department of Computer Science and Engineering,
†Toshiba Corporation, Energy Systems & Solutions Company, Japan

Abstract

Banknotes generally have different designs according to their denominations. Thus, if characteristics of each design can be recognized, they can be used for sorting banknotes according to denominations. Portrait in banknotes is one such characteristic that can be used for classification. A sorting system for banknotes can be designed that recognizes portraits in each banknote and sort it accordingly. In this paper, our aim is to automate the configuration of such a sorting system by automatically detect portraits in sample banknotes, so that it can be quickly deployed in a new target country. We use Convolutional Neural Networks to detect portraits in completely new set of banknotes robust to variation in the ways they are shown, such as the size and the orientation of the face. .

1 Introduction

Quick and accurate banknote sorting is essential in operations that handle large amount of cash such as banks, cash transportation companies, and casinos. Banknote sorting is the task of separating banknotes according to their denominations. In large operations, such a task is largely automated by cash-sorting machines.

Banknote sorting involves tasks such as

- denomination recognition,
- counterfeit detection,
- damaged note detection, and
- counting,

and cash-sorting machines generally perform these tasks automatically and with high throughput.

Cash-sorting machines exist in large numbers for major currencies such as the US Dollar and the Euro. However, there are many more currencies in the world for which such mechanization is less prevalent. The reason is the cost of fine-tuning such machines for banknotes in a given currency. The fine-tuning involves finding and specifying characteristics of each banknote denomination in the given currency so that a pattern-recognition system can detect the characteristics to sort each note according to its denomination. Because each currency has its own design, manual tuning of cash-sorting machine takes time and costs too much in the case of minor currencies.

To alleviate this problem, here we focus on *portraits* that appear on many denominations of various currencies as characteristics to recognize the banknotes. By automatically finding portraits on a given set of sample notes, our automatic configuration system can facilitate the quicker tuning of cash-sorting machines to a new currency.

Although it is relatively easy for humans to distinguish banknotes by viewing the patterns and letters that appear on them, it is inefficient for a machine vision system to try to use the whole face of a banknote to classify it. It is preferable to specify a number of distinguishing characteristic part on each kind of banknote in a given currency for the machine to identify it. For that purpose, pre-identifying such characteristic parts and their positions on the banknote is essential.

Many banknotes feature portraits of various historical figures and prominent individuals. They are portrayed in various ways: sometimes only the head is shown, sometimes from the bust up; they can be a frontal view, in profile, or in between. To detect the portraits on previously unseen set of sample banknotes, a method that is robust to such variety is needed. We use a Convolutional Neural Network (CNN) for that purpose. It is now very well known that CNN shows very high performance in certain recognition tasks, starting with the triumph of the method utilizing it [1] in the Imagenet Large Scale Visual Recognition Challenge (ILSVRC), a contest in general object recognition.

2 Related Work

There are a number of methods known to detect faces using neural networks. For instance, in 1994 Vaillant *et al.* [2] searches for regions that may contain a face in the input image using a neural network, and uses another neural network to determine if it truly contain a face. Also, in 1998 Roley *et al.* [3] proposed to detect faces using a neural network learned using the Bootstrap method, achieving a high performance in detecting frontal faces. This method was later expanded so that it is robust to the rotation of the face [4]. In 2002 Garcia *et al.* [5] presented a neural network that can detect faces in input images of different sizes, with various lighting conditions, and varying orientation of the faces. Further, in 2005 Osadchy *et al.* [6] succeeded in real time face detection and pose estimation using a Convolutional Neural Network.

Not only has neural networks shown their high performance in face detection, but they have demonstrated usefulness in detecting objects other than faces. And in recent years, training deep neural networks has become possible, leading to detection of multiple simultaneous objects. As a newer method to detect multiple classes of objects using neural networks, the R-CNN method [7] by Girshick *et al.* is known. It uses Selective Search [9] to find candidate regions and then classify each region by using a Convolutional Neural Network that has been trained using the ILSVRC 2013 dataset. However, it is not necessarily optimal to use such a method for detecting general objects for our purpose, which is more specialized to finding portraits. Li *et al.* [8] finds that detection of small faces and complicated-looking faces does not always work when a network trained for general object recognition such as used in R-CNN is used. Thus, it is preferable to use a dedicated dataset to train a portrait detection network.

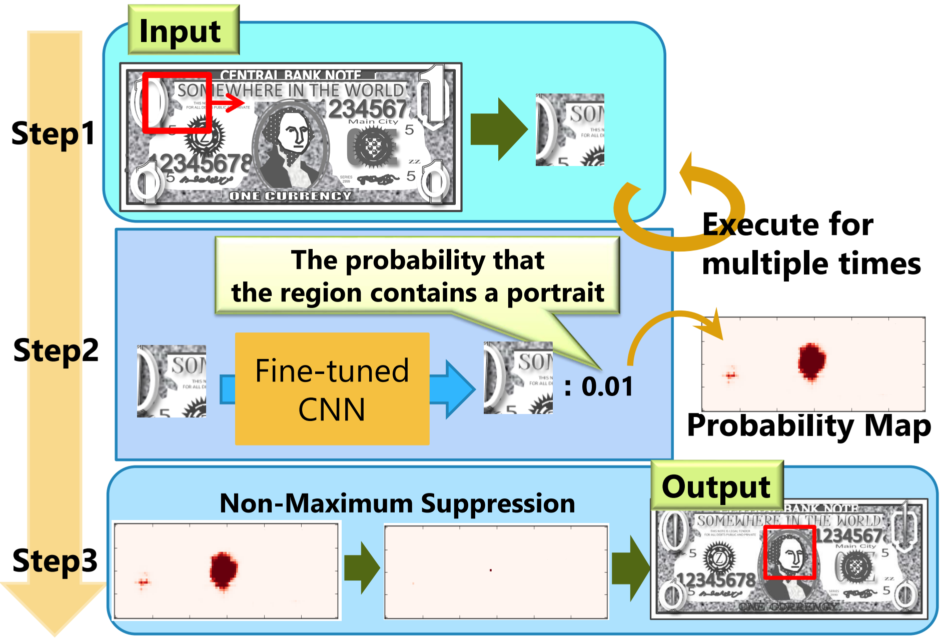

Figure 1. The process of the proposed method. The method determines the rectangles that contain portraits in the input banknote image. Step 1: Candidate region is specified as a moving window. Step 2: Using a CNN, the probability that the candidate region is a portrait is obtained. The probability is recorded as the value of a probability map at the corresponding position. Step 3: A Non-Maximum Suppression process restricts the candidate regions.

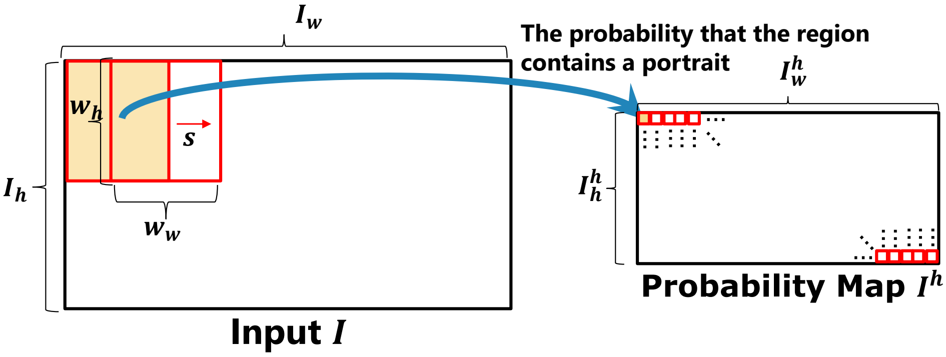

Figure 2. Generating the probability map. The probability that a portrait is in each candidate region specified by a moving window is given by the CNN and assigned to the corresponding position in the probability map. The width of the probability map is the number of the candidate regions when the window is moved in the horizontal direction, whereas its height is the number of the candidate regions when the window is slided in the vertical direction.

3 Proposed Method

Our method takes an image of a banknote and produces the position of rectangles containing portraits in the image, if such exist. If more than one portraits are in the input image, each is detected individually as a separate rectangle. Fig. 1 shows the process of the method. It consists of the following three steps, where Step 3 is done only once, after Steps 1 and 2 are executed for multiple times.

Steps:

1. Candidate region is specified. Here, a moving window generates candidate regions so that all possible candidates are systematically enumerated.
2. Using a CNN, the probability that the candidate region is a portrait is obtained. We construct a map of probabilities by giving the center of the region the probability that it contains a portrait. In Fig. 1, the part with higher probability is shown in red.
3. A Non-Maximum Suppression process restricts the candidate regions.

3.1 Training the CNN

In Step 2 above, a CNN is given a candidate region as the input, and computes the probability that the region contains a portrait. Here, we fine-tuned the Alexnet[1], which is a pre-trained network, using a dataset of portraits we prepared. In fine-tuning it, we re-trained only the two layers on the output side, excluding the three layers after the input layer. This is because not fine-tuning the excluded part does not have much effect on the accuracy as the input-side of a CNN processes relatively local features such as edges. This speed up the training, since there are fewer layers to actually train.

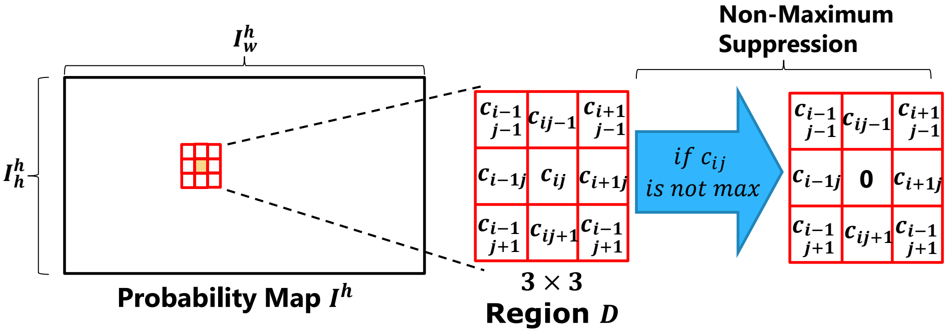

Figure 3. The NMS process. NMS replaces values that are not locally maximum by 0. If the original probability c_{ij} is not maximum among its neighbours, the new value c'_{ij} at the position is set to 0.

3.2 Probability Map

In Step 3, a Non-Maximum Suppression process is used to find the peaks of the probability.. In Fig. 3, the probability map is depicted where higher probability is shown in red regions in the map. In Fig. 2, the process of generating the probability map is depicted. This corresponds to the Steps 1 and 2, where the input image I is turned into the probability map I^h. As shown in this figure, the probability that each candidate region contains a portrait is given as the value of the probability map I^h at the corresponding point in the map, i.e., the center of the region. Thus the size of the probability map I^h is the same as the expanse of the central points of the possible candidate regions. Thus, the width I_w^h and the height I_h^h of the probability map I^h can be computed as:

$$I_w^h = \left\lfloor \frac{I_w - w_w}{s} \right\rfloor + 1, \quad (1)$$

$$I_h^h = \left\lfloor \frac{I_h - w_h}{s} \right\rfloor + 1. \quad (2)$$

Here, s is the stride width.

3.3 Non-Maximum Suppression

Even when there is only a single portrait in the input image, multiple candidate regions that partially overlaps the portrait has non-zero probability of containing the portrait. However, we want to pinpoint the single region that contain the whole of the portrait. Thus, we use a Non-Maximum Suppression (NMS) method to single out the center of such a region for each of (possibly more than one) portraits. In NMS, we look at each 3×3 square in the probability map and if the number at the center is not the maximum among the nine numbers, we replace the number at the center with 0. (The number referred as the nine numbers should be the original one before possibly being replaced by 0 in their turn.) After this process, only the numbers that are higher than their surrounding retain their value. These points are called the peak positions. Thus, applying NMS on the probability map obtained in the previous subsection results in only the points with the highest probability of being the center of a region containing a portrait among their neighbours remaining non-zero. Fig. 3 depicts the NMS process on the probability map. In Fig. 3, c_{ij} denotes the original probability value of I^h at coordinate (i, j) and D the 3×3 region around (i, j). The new value c'_{ij} after NMS can be written:

$$c'_{ij} = \begin{cases} 0 & (c_{ij} = \min_{(l,m)\in D} c_{lm}) \\ c_{ij} & (c_{ij} \neq \min_{(l,m)\in D} c_{lm}) \end{cases} \quad (3)$$

In our method, we additionally smooth the probability map with a Gaussian filter, so that noise is reduced and the peak becomes clearer. Then the remaining non-zero values are peak values, however small. We need to cull small values lest positions with small but peak probability is found to have a portrait. Therefore, we preset a threshold θ and take only the peaks with larger probability than θ as the output points.

4 Dataset

The CNN we use in our method computes the probability that the given input image is a portrait. To train it, we prepared a moderately large number of portrait images appearing in banknotes as positive samples and images of parts of banknotes that do not contain portraits as negative samples. First, we procured 3330 banknote images, each of which shows the whole of a banknote, counting the front and the back side of a banknote as two separate images. Out of these, we randomly sampled 2330 images as the training dataset and used the other 1000 images as test images. Then we manually cut out 889 portraits and 3673 non-portrait regions from the 2330 training images.

5 Experimental Results

5.1 Experiments

We performed portrait detection on the 1000 test images. These images are not used for training. Each shows the whole of a banknote. The moving window has the size $w_w \times w_h$, which is adapted according with the width I_w of the input image as

$$w_w = \lfloor 0.1497 I_w \rfloor, \quad (4)$$

$$w_h = \lfloor 0.4180 I_w \rfloor. \quad (5)$$

Here, the numbers 0.1497 and 0.4180 are the average size of the portraits when the width of the banknote image is set to 1. In the experiments, we varied the stride width s of the moving window as $s = 3, 5, 10, 15$. We also tried two different kernel filter size k and variance σ: $(k, \sigma) = (5, 1.1), (11, 2)$. Furthermore, the value of θ was set to 0.8. Since the input of the CNN is fixed to the input size 227×227 of Alexnet, we resized the candidate region to this size before feeding it to the CNN.

Table 1. Precision and Recall with varying stride s and Gaussian kernel width k

Stride s	3	3	5	5	10	10	15	15
Gaussian kernel width k	5	11	5	11	5	11	5	11
Precision	0.860	0.508	0.600	0.550	0.560	0.597	0.657	0.470
Recall	0.993	0.919	0.946	0.896	0.877	0.561	0.813	0.158

5.2 Evaluation

We calculated how many of the output rectangles correctly detected portraits as the Precision and Recall numbers. We determined if an output rectangle correctly detects a portrait by computing the overlap ratio L of the output rectangle and the ground-truth rectangle. The overlap ratio is defined by the following formula, where A is the area of the ground-truth rectangle, B is the area of the output rectangle, and C is the area of the intersection of the two rectangles:

$$L = \frac{C}{A+B-C}. \tag{6}$$

We determined that the portrait is correctly detected when this number L exceeds a threshold ϕ, and designated the case as the True Positive (TP). Conversely, when L does not exceed ϕ, we determined that the portrait is not correctly detected, and designated the output rectangle as False Positive (FP) and the ground-truth rectangle as Flase Negative (FN). In the experiments, we used the value $\phi = 0.3$, which corresponds to the state when about the half of the rectangles overlap, if the two rectangles are of the same size. Then we counted the number of TP, FP, and FN for each test image. Then we computed the Precision and Recall from the sum of the number of TP, FP, and FN for the 1000 test images, according to the formula:

$$Precision = \frac{TP}{FP+TP}, \tag{7}$$

$$Recall = \frac{TP}{FN+TP}. \tag{8}$$

5.3 Results

Table 1 shows the Precision and Recall with varying stride s and Gaussian kernel width k. It indicates that the smaller the stride s, the better. This is because there are more candidate regions when the stride is smaller, and thus more accurate match between the candidate region and the portrait is possible. Also, it is indicated the smaller of the two Gaussian kernel width produces better results.

6 Conclusion

In this paper, we proposed a method that, given an image of a banknote as the input, find portraits in the input image and output rectangles containing the portraits. We use moving windows to generate candidate regions and compute by a CNN the probability that each region contains a portrait. Then we use Non-Maximum Suppression to output the rectangles with the highest probability of containing portraits. The method achieves high performance in detecting portraits. On the other hand, there are many instances where a portrait is found where there is none. To improve this, we consider the most important avenue is to improve the training dataset, which is not trivial given the limited number of portraits in banknotes.

7 Acknowledgments

This work was partially supported by JST CREST.

References

[1] A. Krizhevsky et al. “Imagenet classification with deep convolutional neural networks,” *Advances in Neural Information Processing Systems*, pp.1097-1105, 2012.

[2] Vaillant et al.: “Original approach for the localisation of objects in images,” *IEE Proceedings-Vision, Image and Signal Processing*, vol.141, no.4, pp.245–250, 1994.

[3] Rowley et al.: “Neural network-based face detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.20, no.1, pp.23–38, 1998.

[4] Rowley et al.: “Rotation invariant neural network-based face detection,” *Proceedings of 1998 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.38–44, 1998.

[5] Garcia et al.: “A neural architecture for fast and robust face detection,” *Proceedings of 16th International Conference on Pattern Recognition*, vol.2, pp.44–47, 2002.

[6] Garcia et al.: “Synergistic face detection and pose estimation with energy-based models,” *The Journal of Machine Learning Research*, vol.8, pp.1197–1215, 2007.

[7] Girshick et al.: “Rich feature hierarchies for accurate object detection and semantic segmentation,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp.580–587, 2014.

[8] Li et al.: “A Convolutional Neural Network Cascade for Face Detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.5325–5334, 2015.

[9] Uijlings et al.: “Selective search for object recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.104, no.2, pp.154–171, 2013.