REPEATED PATTERN EXTRACTION WITH KNOWLEDGE-BASED ATTENTION AND SEMANTIC EMBEDDINGS

Hong QU[1], Yanghong Zhou[1,2], K.P. Chau,[1], P.Y. Mok[1,2,*]
[1]*Institute of Textiles and Clothing, The Hong Kong Polytechnic University, Hunghom, Hong Kong*
[2]*The Hong Kong Polytechnic University Shenzhen Research Institute, Shenzhen, China*
** Corresponding author: tracy.mok@polyu.edu.hk*

ABSTRACT

Repeat pattern is a common structure in textile or graphic design, while scalable and reusable designs are required for production. We propose a new approach, based on Convolutional Neural Network (CNN), for automatic extraction of repeat patterns in this paper. CNNs are good at detecting repeat patterns through extracting multilevel features in the convolutional layers. In our method, we select filters quickly in single images by leveraging the combination of the learned filters from AlexNet's convolutional layers and boundary detection results from VGG16. Moreover, we use template matching to optimize final outputs in order to improve precision. We composed a dataset with 30 images, covering stripe, check and dot distributed patterns, each image with manual ground truth labels. The experimental results show that our method outperforms the state-of-the-art method in both precision and running speeds.

KEYWORDS

repeated pattern extraction, boundary detection, CNN, template matching, textile design, AlexNet

1. INTRODUCTION

Repeat patterns are seen almost everywhere in our daily life, they are used in tiling, wallpaper, textiles, packaging, backdrops for text or illustrations in graphic media, jewelry, and architecture. Designers create them uses a wide range of media in both two and three dimensions. Most modern repeat pattern design is done on computers for guarantee speed and accuracy; electronic repeat patterns have a number of advantages since the artworks are scalable, reusable and can be easily adjusted to fit the screen sizes for production. It usually takes long time to turn a good idea into repeat patterns. Designing completely new patterns may be unrealistic, because the industry currently does not have an efficient mechanism of managing and searching numerous patterns to avoid developing similar products or to reuse existing patterns. Therefore, there is an emerging need for methods to extract repeat patterns automatically and efficiently. In addition, repeat pattern extraction is also required step in many tasks of computer vision and graphics, such as segmenting objects or reconstructing 3D appearance (Lettry *et al.*, 2017).

Automatic repetition detection in a single image is difficult as repeating elements are irregular in either or both their visual contents and spatial positions. Traditional repetition detection methods always first extract feature using handcrafted feature detectors (Liu *et al.*, 2004) and then retrieve similar features by clustering (Guo *et al.*, 2019) or patch correlation (Kuo *et al.*, 2017; Guo *et al.*, 2019) to identify the repeat contents. As the visual variance among semantically similar classes or context and the natural variations (e.g. lighting conditions), the handcrafted feature detector requires to be complex and robust. With the development of deep learning, Lettry *et al.* (2017) and Rodriguez-Pardo *et al.* (2019) proposed to use CNN to extract multilevel features and leverage activations of filters in the convolutional layers to detect repeated patterns on a grid, which shows the good capability of CNN on repetition detection task. However, the method of Lettry *et al.*(2017) have the drawback that the computational cost is expensive to use a large amount of the filters in CNN and use the activation peaks on all filters to seek the regular pattern. To address the problem, Rodriguez-Pardo *et al.* (2019) proposed to abandon the filters that are not significantly activated to reduce the computational cost, but this method made it difficult to evaluate the significance of the filters. They set an activation threshold and only kept the filters with the mean activation greater than the threshold. However, the most activated filters are not necessarily the filters with regular patterns.

In this paper, based on the pre-trained CNNs, we propose to use boundary as *a prior* knowledge to select the possible filters with regular repetitive patterns. As boundary prior is beneficial to screen out most of the background contents and focus on the foreground region, we apply the boundary prior on the filters and select the most activated ones with repetitive patterns to reduce computation. Next, a Hough-like voting method is used to compute the consistent displacement vector. Moreover, a template matching method is also used to optimize the computation of the consistent displacement vector to improve precision. Our main contributions are as follows:

- Combining boundary detection and pre-trained CNN for efficient repetitive pattern detection in a single image.
- Using template matching to optimize repeat pattern detection.
- Experimental results show that our proposed method outperforms Lettry's method (2017) in both precision and running speeds.

2. RELATED WORK

The detection for repeat patterns from a single image has been a challenge task in the field of computer vision. This task requires not only the understanding of image, but also find the regular mechanism automatically from various type of images without principled definition of that mechanism. To solve this problem, researchers have proposed methods in different ways. Liu *et al.* (2004) present a computational model for repeat pattern detection based on the theory of crystallographic groups, thus the repeat pattern classification and detection are based on their geometric structure rule. According to near regular textures, Cai *et al.* (2011) present a general method for detecting repetitive patterns by shape alignment and deformed template matching for the local patches of the texture.

Repeat pattern is of great value in image retrieval, Torii *et al.* (2013) achieve place recognition based on combination of robust detection of repeated image structures and a simple modification of weights in the bag-of-visual-word model. Doubek *et al.* (2010) make a difference is that they are not focusing on detecting repeat pattern on the same image, but on retrieving images of the same object. They implemented the retrieve of buildings on a small dataset.

In fashion industry, both Kuo (2017) and Tao *et al.* (2017) extract repeat pattern for dot distributed pattern. Kuo *et al.* (2017) extract pattern elements' shape information by edge detection firstly. Afterwards, the centroid positions of these identical pattern elements were used to extract the repeat pattern template using a vertical vector method. Tao *et al.* (2017) decompose repeat pattern into the repeat angle and the unit span. Then pattern sliced into pieces on various angles with various spans, and the similarity degree of slices on each angle with each span was calculated to constitute the similarity space in which the peak values suggested the potential repeat angles and unit spans. The similar method is also used in weave pattern repeat detection since fabric surface presents a periodic yarn structure. Zheng (2014) gets corner information by the Harris corner detector firstly, then searches for the adjacent corners in the horizontal or vertical direction to identify the repeat size of the weave pattern. But this method can only process fabric images with clear texture, horizontal weft and vertical wrap. In addition, Guo *et al.* (2019) propose seven local features to define weave pattern repeat, then use the recognition results to judge the fabric structure.

The above methods all define the repeat pattern by extracting different features manually. The experimental results will increase with the number of parameters considered. CNNs are utilized to detecting repeat pattern as it could extract as much information as possible from the available datasets automatically. Lettry *et al.* (2017) are the first to utilize CNN for repeat pattern detection in general situation. In order to make better use of the information captured by each layer in CNN, they sift activation peaks of each filter maps using a standard non-maxima suppression. Then extract the spatial regularities of activation peaks to detect the most consist repeat pattern. Rodriguez-Pardo *et al.* (2019) have made improvements base on Lettry's (2017) work, they add pre-processing part to align image. Besides, they ignore parts of filters which are not activated obviously to speed up. The algorithm suit to detect fronto-planar geometrically regular textures in fast speed. However, the whole paper pays more attention to the image synthesis of the extracted repeat patterns, so our experimental results are not compared with them.

3. METHOD

3.1 System Overview

CNN network has shown great capability of image representation learning in many tasks, such as image classification (Liu and Deng, 2015), object detection (Cai *et al.*, 2016), semantic segmentation (Kampffmeyer *et al.*, 2016). The CNN network consists of a set of layers, convolutional filter layer, pooling layer, fully connected layers, and so forth. The filters of different layers learn different semantic features at varied scale levels. Information learned from the higher layer contains more semantic information than the lower layers. Hence, when an image input to a CNN, the filters generate feature maps that respond to various semantic and scale levels, as well as region localization. When a repeated pattern exists in the input image, the feature maps that activate the repeated pattern are generated by CNN filters and the activated region follow a regular pattern. In this paper, we propose to fuse a boundary detection CNN and a classification CNN to select the most activated filters with repetitive patterns. Next, we compute the most activated peaks and consistent displacement vector using the selected filters. Finally, an optimization algorithm is proposed to refine the consistent displacement vector. The overall framework is shown in Figure 1.

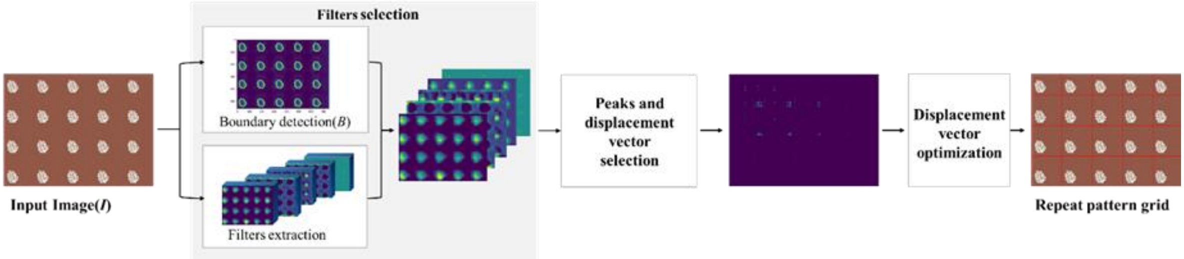

Figure 1. Illustration of our method: step (1) is filter selection; step (2) peaks and displacement vector selection; and step (3) optimization.

3.2 Filter selection

Given a classification CNN, we denote the filters in CNN as $F = \{F_n | n = 1,2 \dots, N\}$, $F_n = \left\{f_{l_n} \middle| l \in L_n\right\}$ where L_n is the set of convolutional filters of the n-th layer. When the input image I containing a repeat pattern passes over the CNN network, CNN filters will generate characteristic regularly activation feature maps. The goal of the first step is to select filters with the possible regular activated feature maps. As boundary detection can greatly reduce the amount of image data and remove weakly related information while retaining important structural attributes, we use boundary detection as *a prior* to assist the filter selection. For boundary detection, we used the RCF network (Liu *et al.*, 2019), which could produce side outputs of each inside layers based on VGG16 (Simonyan and Zisserman, 2014). Due to that deeper networks hardly avoid problems such as vanishing/exploding gradients and training data shortage, RCF network uses the CNN features of all the conv layers directly to perform the pixel-wise prediction. Therefore RCF network could obtain accurate representations for objects or object parts in different scales.

Let us denote the detected boundary as B, we multiply B with the activated feature maps $f_{l_n}(I)$ of each layer and sum over all pixels：

$$f_{l_n}^B = \sum_{i,j} B \cdot f_{l_n}(I) \tag{1}$$

where (i, j) represent the pixel, the most possible filter with regular activated feature maps l_n^* of the n-th layer can be obtained by

$$l_n^* = \underset{l \in L^n}{\operatorname{argmax}}(f_{l_n}^B) \tag{2}$$

3.3 Peaks and displacement vector selection

With the selected filters $\{f_{l_n^*} | n = 1,2, \dots, N\}$, we extract the activation peaks of feature maps and compute the probable size of repeated elements. For each filter $f_{l_n^*}$, we use the standard non-maxima suppression (NMS) algorithm to extract local maximum peaks $P_{f_n}^*$ in a region of certain distance. When the minimum neighborhood distance is $dist_{\min}$, $P_{f_n}^*$ can be computed in the region of $2dist_{\min} + 1$ using NMS algorithm. Due to different resolutions and strides of the filters, $dist_{\min}$ is determined by the number of local maximum peaks to guarantee enough extracted peaks on each filter map. We set the minimum number of extracted peaks as n, if the number of generated peaks is less than $n_{\min}$, then $\mathrm{dist}_{\min} = \mathrm{dist}_{\min} - 1$, loop recursively until the number of generated peaks is more than $n_{\min}$. The algorithm is shown below:

Input: the filter $f_{l_n^*}$, the required number of extracted peaks $n_{\min}$, the initial minimum neighborhood distance $dist_{\min}$.

```
$P_{f_n}^*$=NMS ($f_{l_n^*}$, $dist_{\min}$)
If length ($P_{f_n}^*$) >$n_{\min}$:
    Return $P_{f_n}^*$
Else:
    $dist_{\min} = dist_{\min} - 1$
    $P_{f_n}^*$=NMS ($f_{l_n^*}$, $dist_{\min}$)
    Return $P_{f_n}^*$
Output: $P_{f_n}^*$
```

For each selected filter $f_{l_n^*}$, we extract the activation peaks $P_{f_n}^*$ using the above algorithm. We then use the activation peaks of all the filters to compute the probable size of repeated pattern by the Hough-like voting algorithm for consistent displacement vector selection (Lettry *et al.*, 2017). We form a set $D_{f_{l_n}^*}$ of displacement vectors by computing a vector for each pair of peak activations in each filter $D_{f_{l_n}^*} = \left\{d^{i,j}: |p^i - p^j|, \forall p^i, p^j \in P_{f_n}^*, i \neq j\right\}$, where $|.|$ denotes the element-wise absolute value on vectors. Then displacement vectors for selected filters in each layer cast votes into the displacement vector Hough voting space $V: R^2 \to R$. We borrowed Lettry's (2017) method using:

$$V = \sum_{x \in R^2} \sum_{\substack{l \in L \\ f_{l_n^*} \in F_n}} \frac{1}{|D_{f_{l_n}^*}|} \sum_{d^{i,j} \in D_{f_{l_n}^*}} V_{f_{l_n}^*, i, j} \qquad (3$$

where V is smoothed using a 2-dimensional Gaussian filter with a kernel size of $\begin{pmatrix} \sigma_{L_n}^2 & 0 \\ 0 & \sigma_{L_n}^2 \end{pmatrix}$. The optimal displacement vector d^* , which has the dimensions of the minimal repeat pattern, is extracted as the maxima of voting space on the x and y axes :

$$d^* = (\operatorname*{argmax}_x V_{x,0} , \operatorname*{argmax}_y V_{o,y}) \qquad (4)$$

3.4 Displacement vector optimization

When we know the size of the repeat pattern, repeat patterns of the same size can be found at any starting point in the same picture (see Figure 2). It is also important to note that when the starting point is known and the size of the repeat pattern d^* is accurate, the image region of all the grids are the same. However, d^* may be inaccurate, which will result in different grid regions. To refine the size of repeat pattern d^*, we use template matching.

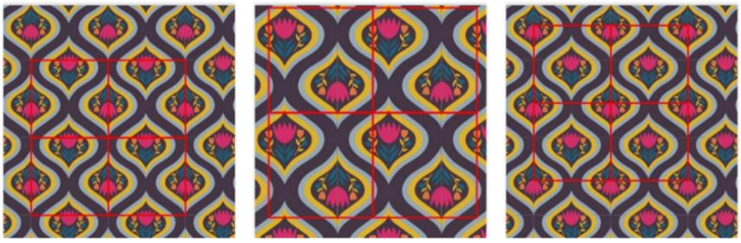

Figure 2. Examples of repeat pattern: The left two images show repeat patterns of different scales. The first image and third image show repeat patterns of the same size can be found at any starting position.

Let us denote the starting point of repeated pattern as $o = (0,0)$. Given an image I, we crop the region from the starting point o with size of d^* and use the region as a template of repeat pattern T. Given the template T, the matching process overplays the template on the searched image (W * H), moves it to all possible areas and computes a numerical index that indicates how well the template matches the image in that area. We denote the area on the searched image covered by the template as S_{hk}, where $1 \le h \le W - d_x^*, 1 \le k \le H - d_y^*$. The correlation is used to measure the matching degree of template T and S_{hk}:

$$cor = \frac{\sum_{h,k}(T(h,k) \cdot S(h,k))}{\sqrt{\sum_{h,k} T(h,k)^2 \cdot \sum_{h,k} S(h,k)^2}} \tag{5}$$

The value of cor is between -1 and +1, with larger values representing a stronger relationship between the two images.

It should be noted that the template is not searched on the whole image, to reduce the computation time, but on the cropped images from the starting point o with size (d_x^*, H) and size (W, d_y^*) which are denoted as I_{ox} and I_{oy} respectively. To refine the d_x^*, the template moves on the cropped image I_{ox} and search the matched areas S_{h1} along y axis. From the starting point ($\frac{1}{3} d_x^*$,0), as long as the matched area S_{h^*1} can be found on the cropped image I_{ox}, d_x^{**} is refined to $h^* + \frac{1}{3} d_x^*$. Similarly, from the starting point $(0, \frac{1}{3} d_y^*)$, moving the template on the cropped image I_{oy} along the y axis, matched area S_{1k^*} can be founded on I_{oy} then d_y^{**} is refined to $k^* + \frac{1}{3} d_y^*$. Considering the pixel error that makes the templates of adjacent pixels similar, we also compute the distance Δd between then template S_{h1} and next template S_{h2}. If $\Delta d < 2$, then $d_x^{**} = d_x^{**} + \Delta d$. The same operation until the distance $\Delta d \ge 2$. This operation is not only useful to avoid the pixel error, but also is able to identify the stripe direction on the strip image. Because on the strip image, when searching template along the reverse direction of stripes, we can find the matched template on every pixel and obtain that $d_x^{**} = w - d_x^*$ or $d_y^{**} = h - d_y^*$.

Taking the refinement of d_x^* as an example, we crop input image I from the starting point (0,0) with size of (d_x^*, H) and search the matched areas with template T, which is shown in Figure 3(a). As existing the matched area S_{h^*1}so that $h^* > \frac{1}{2} d_x^*$, the $d_x^* = h^*$. The grid drawing using the refined d_x^* and the computed start point o^* is shown in Figure 3(b), which can be seen that the detected pattern is much more accurate than before.

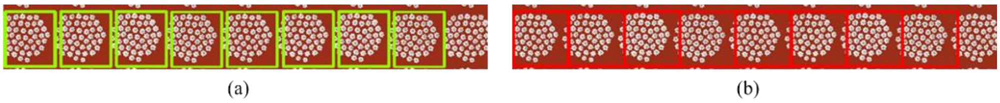

Figure 3. The optimization effect of x-axis. (a): Effect before optimization. (b): Effect after optimization.

4. EXPERIMENTAL RESULTS AND DISCUSSION

We present qualitative and quantitative results of our algorithm and compare to the state-of-the-art method in this section. To allow for quantitative evaluation, we first introduce our dataset with manual ground truth labels.

4.1 Dataset preparation

Graphic patterns are basically classified into dot distributed patterns, check patterns and continuous stripe patterns (Kuo *et al.*, 2017; Tao *et al.*, 2017). There is no public dataset focusing on repeat patterns, and some downloadable images have either no ground truths (repeat pattern size) or the images have less than two complete repeat patterns, which would affect the analysis. Therefore, we created a new dataset based on some vector artwork downloaded from WGSN. We downloaded 30 self-created images covering three categories: stripe pattern, check pattern and dot distributed pattern. The image resolution is around 500*500pixel. There are 10 images for each type of pattern and more than two repeat patterns lying on a regular grid on the image. We manually defined the minimum repeat pattern size on the images as the ground truth.

4.2 Evaluation metrics

As mentioned in Section 3.4, the position of a repeat pattern in an image is subjective. When an image composed of repeat patterns, a pattern of the repeat size selected from any position as the starting point could obtain the same density effect, as shown in Figure 2. Thus, we assume that each image starts draw grid from the same starting point (0,0) for unified evaluation. Intersection-over-Union (IoU) is used to evaluate the performance of repeated pattern detection. Since the repeat pattern has the same start point, d_x^* and d_y^* are the size of estimated repeat pattern in x-axis and y-axis respectively, g_x and g_y are the ground truth of minimal repeat pattern size, IoU accuracy is computed as follows:

$$IoU = \frac{\min(d_x^*,g_x)*\min(d_y^*,g_y)}{d_x^*,*d_y^*+g_x*g_y-\min(d_x^*,g_x)*\min(d_y^*,g_y)} \tag{6}$$

In particular, when the image is stripe, the predicted and ground truth values of the reverse direction are equal to the corresponding image side length.

4.3 Qualitative evaluation

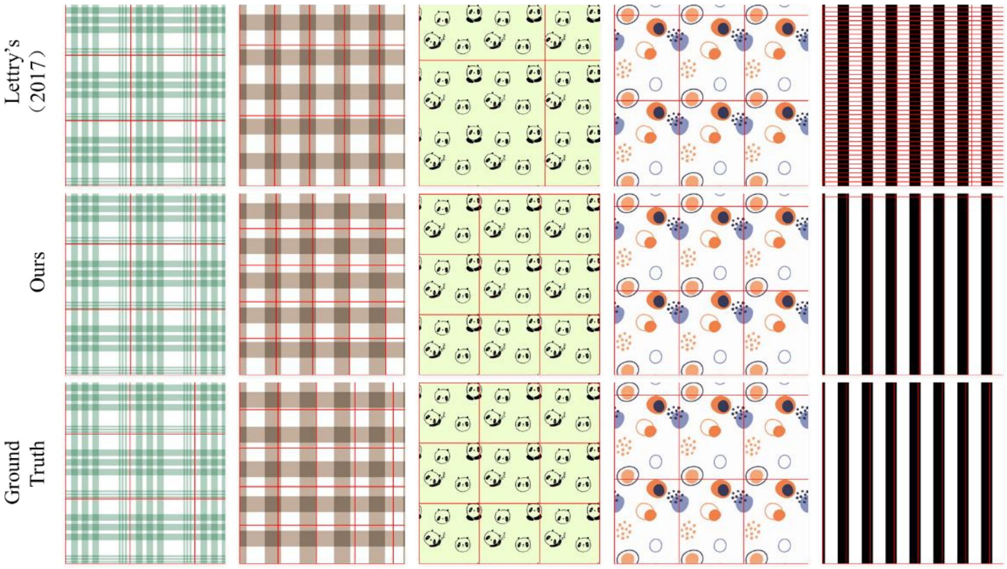

Figure 4. Qualitative comparison to related work: Lettry *et al.* (2017) and our ground truth. Our method performs better on all examples.

We present the repeat patterns detected by our algorithm and Lettry's (2017) method on different types of images of the dataset in Figure 4. As shown, our method performs more precise on the detected size of repeat pattern than Lettry's (2017) method. Our method also performs well on complex patterns. As shown in the fourth column, Lettry's (2017) method corrupted due to the spatial non-regularity in the pattern combined with many pattern elements, but our method found the repeat pattern successfully. Moreover, for the repeat pattern detection on stripe image (column 5 in Figure 4), with the help of the optimization using template matching, our method can identify the strip direction. These results show that our method is very accurate on repeat pattern detection, performs well even on complex pattern images, and our method is able to identify the strip direction if stripe images.

4.4 Quantitative evaluation

The dataset with ground truth was used for quantitative evaluation: we computed average IoU, the accuracy at different thresholds of IoU, and the time costs over the dataset. We compare our method with Lettry’s (2017). Table 1 summarizes the quantitative evaluation of two methods over different types of our dataset: stripe, check, and dot distributed patterns.

Table 1. Experimental results

	Lettry's (2017)				Our Method			
	IoU Average	IoU ≥0.8 prec./(%)	IoU ≥0.9 prec./(%)	Time Costs/(s)	IoU Average	IoU ≥0.8 prec./(%)	IoU ≥0.9 prec./(%)	Time Costs/(s)
Check	0.88	80.00	60.00	4060	0.88	90.00	70.00	33
Dot	0.81	80.00	60.00	3989	0.94	90.00	90.00	19
Stripe	0.79	70.00	70.00	3380	0.95	90.00	90.00	127
All images	0.83	76.67	63.33	3810	0.93	90.00	83.33	60

Computer config: CPU Intel i7-6700x, GPU NVIDIA GTX 1080 Ti, Memory 32GB.

It can be seen from the table that average IoU for Lettry’s (2017) and our method is 0.83 and 0.93, respectively, which demonstrates the benefits of our method. When the threshold of IoU is set to 0.8, the precision of our method is 90.00%; when the threshold is 0.9, our precision (83.33%) is 20% higher than theirs. Among the three types of pattern, the IoU precision of our method is higher than Lettry’s (2017) especially in dot and stripe patterns under different thresholds. For stripe patterns, the precision for our method and theirs are 90% and 70%, respectively, when the threshold is 0.8; the precision is the same when the threshold was 0.9.

Comparing the time costs, our proposed method demonstrates the combination of boundary detection and learned filters could reduce computation and save time. For the same stripe pattern image, our method takes about 2 minutes, while Lettry's (2017) method takes about 1 hour. On the pattern of check and dot, the efficiency of our method is more obvious, which we need 100 times less computation time cost. Since the stripe pattern circulates in one direction, after filter selection it extracts more peaks in the reverse direction when extracting peaks of filter maps and costing more time at Hough-like voting process. It could explain the longer time costs of stripe patterns by our method, as shown in Table 1.

5. CONCLUSION

We presented a new method for repeat pattern detection through the boundary combination and template matching optimization. To reduce the computation, we use the detected boundary to select the most possible filters with repeat patterns on multi-level layers. Then, an optimized peak extraction algorithm and Hough voting algorithm are used to extract peaks on the selected filters and compute the repeat pattern size. After the repeat pattern size is obtained, a template matching method is used to optimize it over the whole image. Experimental results show that our proposed method obtain better results both in terms of precision and speed. In particular, our method increases computational speed by more than 60 times on average. In the future, we will explore using texture classification model and feature matching method to further improve the overall accuracy, efficiency and robustness of the method.

ACKNOWLEDGMENT

The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong

Special Administrative Region, China (Project No. 152161/17E and 152112/19E). The work was also partially supported by The Innovation and Technology Fund (Grant No. ITP/013/18TI) and Shenzhen Science and Technology Innovation Commission (Project No. JCYJ20170303160155330).

REFERENCES

CAI, Y. & BACIU, G., 2011. Detection of repetitive patterns in near regular texture images. *2011 IEEE 10th IVMSP Workshop: Perception and Visual Signal Analysis*, NY, USA, pp. 60-65.

DOUBEK, P. et al, 2010. Image matching and retrieval by repetitive patterns. *2010 20th International Conference on Pattern Recognition*, Istanbul, Turkey, pp. 3195-3198.

GUO, Y. et al, 2019. Automatic recognition method for the repeat size of a weave pattern on a woven fabric image. *Textile Research Journal,* Vol. 89**,** No.14, pp.2754-2775.

KUO, C.-F. J. et al, 2017. Image database of printed fabric with repeating dot patterns part (I)–image archiving. *Textile Research Journal,* Vol. 87**,** No.17**,** pp. 2089-2105.

LETTRY, L. et al, 2017. Repeated pattern detection using CNN activations. *2017 IEEE Winter Conference on Applications of Computer Vision (WACV),* CA, USA, pp. 47-55.

LIU, Y. et al, 2004. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Transactionas On Pattern Analysis And Matchine Intelligence,* Vol. 26**,** pp. 354-371.

RODRIGUEZ-PARDO, C. et al, 2019. Automatic extraction and synthesis of regular repeatable patterns, *Computers & Graphic,* Vol.83**,** 33-41.

SIMONYAN, K. & ZISSERMAN, A., 2015. Very deep convolutional networks for large-scale image recognition. *ICLR*.

TAO, C., ZHOU, J. & YIN, M. J. T. R. J. 2017. Automatic identification of textile pattern consecutiveness based on similarity space. *Textile Research Journal,* Vol. 87, No.2, pp. 224-231.

TORII, A. et al, 2013. Visual place recognition with repetitive structures. *Proceedings of the IEEE conference on computer vision and pattern recognition,* Oregon, USA, pp.883-890.

ZHENG, D., 2014. A new method for the detection and classification of weave pattern repeat. *Textile Research Journal,* Vol. 84**,** No.15, 1586-1599.

Liu, Y. et al, 2019. Richer convolutional features for edge detection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* Vol. 41, NO. 8. pp. 1939-1946.

CAI, Z. et al, 2016. A unified multi-scale deep convolutional neural network for fast object detection. *European conference on computer vision, Springer,* 354-370.

KAMPFFMEYER, M. et al, 2016. Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition workshops,* pp. 1-9.

LIU, S. & DENG, W., 2015.Very deep convolutional neural network based image classification using small training sample size. *2015 3rd IAPR Asian conference on pattern recognition (ACPR), IEEE,* pp. 730-734.