signals

MDPI

Article

Efficient and Effective Detection of Repeated Pattern from Fronto-Parallel Images with Unknown Visual Contents

Hong Qu [1], Yanghong Zhou [1,2], P. Y. Mok [1,2,*], Gerhard Flatz [3] and Li Li [4,*]

[1] School of Fashion and Textiles, The Hong Kong Polytechnic University, Hong Kong; hong.qu@connect.polyu.hk (H.Q.); yanghong.zhou@connect.polyu.hk (Y.Z.)
[2] Research Centre of Textiles for Future Fashion, The Hong Kong Polytechnic University, Hong Kong
[3] KTC Limited, Hong Kong
[4] Division of Integrated Systems and Design, The Hong Kong University of Science and Technology, Hong Kong
* Correspondence: tracy.mok@polyu.edu.hk (P.Y.M.); lillyli@ust.hk (L.L.)

Abstract: The effective detection of repeated patterns from inputs of unknown fronto-parallel images is an important computer vision task that supports many real-world applications, such as image retrieval, synthesis, and texture analysis. A repeated pattern is defined as the smallest unit capable of tiling the entire image, representing its primary structural and visual information. In this paper, a *hybrid method* is proposed, overcoming the drawbacks of both traditional and existing deep learning-based approaches. The new method leverages deep features from a pre-trained Convolutional Neural Network (CNN) to estimate initial repeated pattern sizes and refines them using a dynamic autocorrelation algorithm. Comprehensive experiments are conducted on a new dataset of fronto-parallel textile images as well as another set of real-world non-textile images to demonstrate the superiority of the proposed method. The accuracy of the proposed method is 67.3%, which represents 20% higher than the baseline method, and the time cost is only 11% of the baseline. The proposed method has been applied and contributed to textile design, and it can be adapted to other applications.

Keywords: repeated pattern; fronto-parallel images; convolutional neural network (CNN); deep feature selection; template matching

Academic Editor: Ignacio Bosch Roig

Received: 5 December 2024
Revised: 22 January 2025
Accepted: 23 January 2025
Published: 24 January 2025

Citation: Qu, H.; Zhou, Y.; Mok, P.Y.; Flatz, G.; Li, L. Efficient and Effective Detection of Repeated Pattern from Fronto-Parallel Images with Unknown Visual Contents. *Signals* **2025**, *6*, 4. https://doi.org/10.3390/signals6010004

1. Introduction

Repeated patterns can be found everywhere, not only in everyday environments, such as on textiles, packaging, buildings, and architectural structures, but also within molecular structures and biological processes that are invisible to the naked eye. A repeated pattern is defined as the minimum representative unit (or tile) of an image, which can replicate the image's original appearance when tiled [1,2]. Detection of repeated patterns from input images is a fundamental computer vision problem, which serves as a key enabler for diverse real-world applications. For example, in the textile industry, repeated pattern detection automates quality control, defect detection, and assists design element extraction for resource preparation, significantly reducing the reliance on manual labor. Similarly, in architecture, detecting repeating patterns in buildings supports place recognition, urban planning, and 3D modeling. Repeated patterns differ considerably in terms of *image content* and *topological regularity*. Figure 1 shows examples of repeated patterns with distinct content and topological regularity. Specifically for textile images, they often exhibit regular or near-regular topological structures but feature highly diverse visual content, making them a unique and challenging domain for repeated pattern detection.

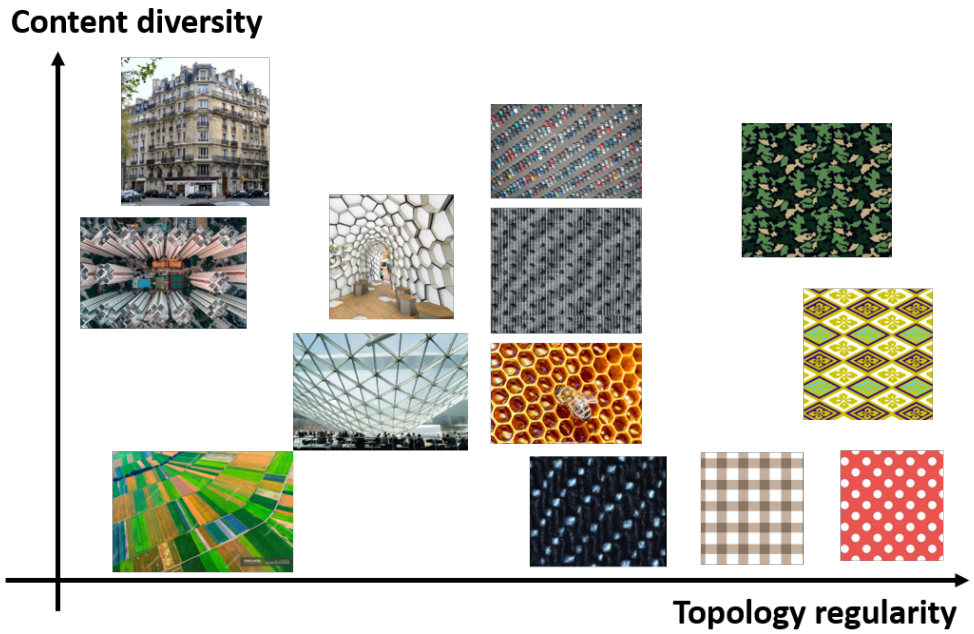

Figure 1. Repeated patterns vary in terms of content diversity and topology regularity.

Recognizing recurring patterns is practically instinctive for humans, but it is difficult for computers due to the wide variety of image types and the complexities introduced by perspective distortions, lighting variations, occlusions, and so forth. Over the years, many methods or algorithms have been proposed for repeated pattern detection. Earlier approaches, such as template matching and signal processing methods, were computationally efficient and effective for regular and noise-free patterns [3,4]. However, these methods often fail when faced with structural variations caused by occlusion, projection distortions, or deformations. To overcome these limitations, feature-based methods were developed, which abstract input images into a set of local features (e.g., keypoints or descriptors) and detect repeated patterns by identifying repetition rules among these features [5–8]. While feature-based methods offer improved adaptability to structural variations, they rely heavily on handcrafted features, such as Scale-Invariant Feature Transform (SIFT) [7] or a Histogram of Oriented Gradients (HOGs) [8], which have limited capacity to handle diverse content and are prone to failure in dynamic or irregular scenarios. In recent decades, Convolutional Neural Networks (CNNs) have revolutionized computer vision by enabling robust feature extraction at multiple levels of abstraction. CNNs have shown great performance in tasks such as image classification [9], signal processing [10], and object detection [11]. For repeated pattern detection, CNNs have been successfully applied to extract hierarchical features for pattern size prediction and position estimation [2,12]. Unlike traditional handcrafted descriptors, pre-trained CNNs can generalize effectively to new tasks without requiring large task-specific datasets, which makes them particularly attractive for repeated pattern detection. However, CNN-based methods still face three critical challenges. First, the computational cost of CNNs scales rapidly with image size, making them unsuitable for high-resolution or complex images. Second, although CNNs improve robustness to diverse content, achieving an optimal trade-off between accuracy and efficiency remains an open question. Third, CNN-based methods often lack the ability to refine predictions dynamically to adapt to irregular patterns or noisy data, limiting their applicability in real-world scenarios.

Regardless of the approaches, repeated pattern detection methods usually cover two key components: (1) feature extraction and processing and (2) prediction model optimization. In principle, adjusting the coupling levels of these two key components can

result in methods that are able to detect repeated patterns from any images, namely images with various combinations of the two complexity dimensions defined in Figure 1, e.g., with similar content but varying topological structures or with similar topology structure but varying contents. This study proposes a *general-purpose hybrid method* that combines the strengths of deep feature extraction with an autocorrelation-based optimization model, overcoming the limitations of existing methods. The method is designed to detect repeated patterns robustly and efficiently across a wide range of image types, including those with diverse content and structural complexity. Specifically, the proposed method leverages pre-trained CNNs for efficient feature extraction, avoiding the need to train a dedicated model and ensuring scalability across various image types. The method also introduces an autocorrelation-based optimization algorithm to dynamically refine the detection of pattern sizes and positions, adapting to irregular or noisy data. Additionally, the method incorporates a feature selection mechanism based on edge detection to extract a small, informative subset of feature maps, significantly reducing computational cost without compromising accuracy.

In order to demonstrate the practical value of the proposed method, the method was applied to detect repeated patterns from textile images. To this end, a dataset of 841 textile images with manually annotated ground truth was developed in this study that provides a reliable benchmark for repeated pattern detection research. This dataset reflects the inherent complexity of textile patterns, characterized by diverse visual contents and near-regular structures, making it an ideal testbed for validating the proposed method's performance.

The main contributions are summarized as follows:

- It introduces the first hybrid method that combines deep feature extraction with an autocorrelation-based optimization model, achieving a unique balance of robustness, speed, and adaptability.
- A feature selection mechanism based on edge detection is introduced, which significantly reduces computational cost while maintaining accuracy.
- An autocorrelation model with a dynamic similarity threshold is designed to efficiently refine repeated pattern sizes, adapting to variations in image complexity and irregularity.
- A dataset of 841 textile images with manually annotated ground truth is developed, providing a reliable benchmark for repeated pattern detection research.
- Extensive experiments demonstrate the superior accuracy and efficiency of the proposed method compared to state-of-the-art (SOTA) approaches, with ablation studies validating the effectiveness of individual components.

Although textile images, mostly with near fronto-parallel repeated patterns of high resolutions and varying contents, serve as an excellent representative testbed, the proposed method is not limited to these type of images only. The proposed method can adapt to other types of repeated patterns with irregular structures by adjusting the coupling between the relevant modules designed for (1) feature extraction and processing and (2) prediction model optimization.

2. Related Work

Existing repeated pattern detection methods vary in terms of either (1) prediction model optimization or (2) feature extraction and processing approach. These works are reviewed as follows.

2.1. Prediction Model Optimization

When an image consists of repeated patterns with a regular layout and very similar content, the locations and sizes of the repeated patterns can be detected by autocorrelation

models or related algorithms [3,13–15]. Chen et al. [3] used Fourier transform to build a similarity space to identify repeated patterns in textile images. Some other researchers leveraged template matching [16] for repeated pattern detection. For example, Qayum and Naseer [17] selected the initial 10 or 100 columns of pixels from the query image as the template patch. Their method considered that the size of the repeated pattern was equal to the distance between the starting point on the condition that the repeating unit should be precisely the same. Similar approaches include Neupane et al. [15] and Lin et al. [14]. Although these methods work very efficiently for images of any intensity with strong or weak features, a number of limitations need to be noted. On one hand, they need a given template patch in advance [18,19], and the size deviation between the given template patch and the actual repeated pattern will affect the accuracy of the final results. On the other hand, they are poor in robustness and have a low tolerance for noise. Thus, they cannot achieve good detection results on repeated patterns with variations, e.g., by occlusion or other forms of deformation.

2.2. Feature Extraction and Processing

2.2.1. Local Feature-Based

Since image descriptors represent image information such as textures and shapes, the repeated patterns in an image can be viewed as a set of sparse repeated features whose location information is consistent with each other. The repetition structure is extracted by hypothesizing links between the repeated feature points [18,20–22]. Some algorithms use SIFT [7] as feature descriptors [5,23,24]. However, these methods are only applicable to images with distinct corner or edge features and simple content. Furthermore, some feature descriptors are handcrafted and designed based on professional knowledge in specific fields [25]. Therefore, even though local feature-based approaches are widely used in two-dimensional images, they are complex in feature descriptors' design and hard to extend to other domains of images. Additionally, many algorithms require image rectification as a pre-processing step to ensure accuracy and better applicability [26].

2.2.2. Deep Feature-Based

A trained CNN contains a set of convolutional layers with fixed weights that can capture the hierarchical information of an image. A convolutional *filter* in the convolutional layer generates different activations of the input image as pre-layer outputs and encodes the spatial information of discriminative regions into a *feature map* [27,28]. A feature map may alternatively be thought of as detection scores acquired after applying a filter over particular spatial places in a 2D image; the activation value at the i-th location quantifies the significance of the pixel there [29]. In CNNs, lower convolutional layer feature maps contain local information such as edge, line, and corner, while filters in deeper layers take the outputs of lower layers as input and generate feature maps that contain more semantic information [30,31]. Therefore, in many tasks, a trained CNN can be treated as a more robust feature descriptor and replace local image feature descriptors. For example, El Amin et al. [32] used CNNs' deep features for change detection in satellite images. Xiang et al. [33] leveraged deep features to help retrieve similar fabric images. Wang et al. [34] used deep features for RGB-D image tracking.

The potential of a deep feature-based approach in repeated pattern detection was presented by Lettry et al. [12]. The authors demonstrated that repeated patterns' key points in two-dimensional images have spatial consistency with certain activations within feature maps; hence, extracting activations can assist in repeated pattern detection. Nevertheless, their method took almost all the feature maps of a trained CNN model into consideration for repeated pattern detection, which is computationally expensive and results in poor

efficiency. In another related study, Rodriguez-Pardo et al. [2] reduced computation by selecting filters (feature maps) that are active more than others. A similar idea of data reduction was used in landscape image analysis: Malik and Robertson [29] used principal component analysis (PCA) to compress the CNN feature maps to a compact representation that best encodes landscape image information for landscape similarity analysis.

Overall, these cases support the view that deep-feature-based methods are robust but not concerned with the accuracy of the repeat pattern size. On the contrary, the approaches in use for autocorrelation models are accurate, efficient, yet noise-sensitive.

3. Method

To achieve a good trade-off between accuracy, robustness, and efficiency for repeated pattern detection in an unknown image, a hybrid method by combining deep features extraction and processing with an autocorrelation model is proposed in this paper. The overall pipeline of the proposed method consists of five steps, where the first two steps of (a) and (b) are related to *deep feature extraction and processing*, and steps (c) to (e) are for *repeat size prediction and optimization* (see Figure 2). The CNN has shown a great capability for representation learning and feature extraction, and the filters of different convolutional layers capture spatial information of discriminative regions at varying scales and encode them in feature maps. As shown in Figure 2, the input to the proposed method is an unknown image, and the output is the image with a repeated pattern grid. The detail of the proposed method is described in the following.

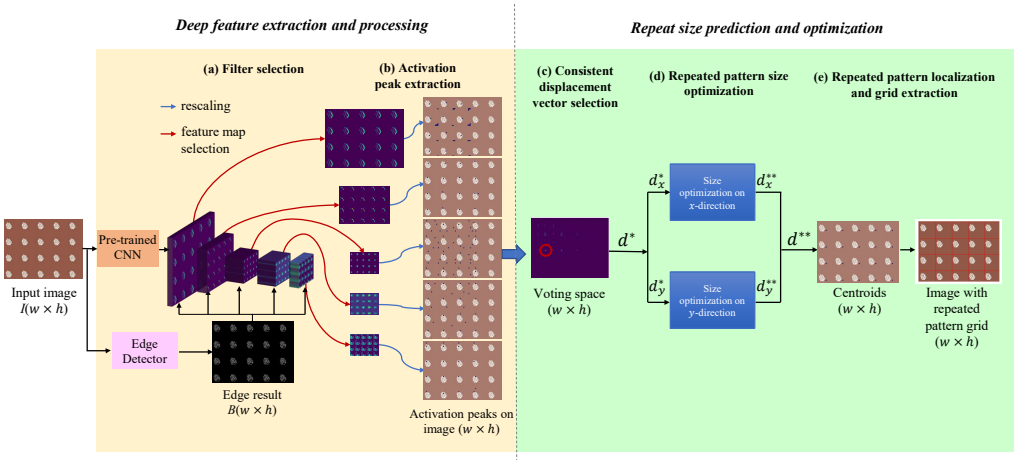

Figure 2. Illustration of the proposed method for efficient repeated pattern detection.

3.1. Filter Selection

In every image representation scheme, selecting appropriate features is an essential step because it enables feature dimension reduction and improves the efficiency of feature processing [35,36]. As discussed, a CNN can generate an excessive number of feature maps. For example, the classic CNN model, AlexNet, has five convolutional layers, with corresponding 96, 256, 384, 384, and 256 filters within these layers [37]; when an image passes AlexNet, it will generate 1376 $(= 96 + 256 + 384 + 384 + 256)$ feature maps. When all these 1376 feature maps are input to a prediction model for repeated pattern defection, the analysis is computationally expensive, but may not be necessary because not all feature maps carry equally important information for the said task. Filter selection techniques are therefore used to determine an appropriate subset of feature maps to adequately represent input images. This appropriate subset of feature maps needs to be both concise and take hierarchical information into account.

Different from previous studies, using PCA for feature map selection, edge information is proposed for filter selection in this study. This is inspired by the fact that edge detection extracts object boundaries and visually salient edges that persevere the gist of the image and ignore weakly related information [38,39]. Moreover, the edge detection result is binary, which makes it computationally efficient and straightforward to combine with feature maps. Specifically, the binary edge map ($[0, 1]$) can directly multiply with the feature maps to extract rich and informative features. Thus, image edge detection results are leveraged to select filters and their corresponding feature maps in the current method.

Given a pre-trained classification CNN, the convolutional filters in the CNN are denoted as $F = \{F_n | n \in N\}$, $f_n \in F_n$, where N is a set of convolutional layers, and F_n is the set of convolutional filters in the n-th layer. f_n is one of the filters of the n-th layer, $f_n \in F_n$. Use the same example of AlexNet as an illustration; it has 5 convolutional layers, thus $N = 5$ and F_1 are a set of 96 filters in the first layer, F_2 is a set of 256 filters in the second layer, and so forth. When an input image I with size $w \times h$ passes through AlexNet, the convolutional filters will generate 1376 activated feature maps containing topology layout information of its repeated patterns.

Assume B denotes the edge result for image I, which can be obtained by any edge detector, e.g., the Canny detector [40], and B is a binary matrix with the same size of $w \times h$ in $[0, 1]$. The edge result B is used to select the filter that captures the most significant features for each convolution layer. To do so, B is first resized to B_n, the same size of the feature maps $f_{n,m}(I)$, for corresponding convolutional layer n. The resized B_n is then multiplied with each of the M feature maps $f_{n,m}(I)$ of the n-th layer as follows:

$$f_{n,m}^{B} = B_n \cdot f_{n,m}(I) \tag{1}$$

where $\cdot$ represent the dot product. An activation map f_n^* is calculated among M feature maps of regular activated feature by

$$f_n^* = \arg\max\left(f_{n,m}^{B}\right) \tag{2}$$

It is worth emphasizing that only one representative filter is chosen in each convolutional layer. An example of selected filters is shown in Figure 2a, which was obtained using a classic Canny edge detector [40]. Later in Section 5, a comparative study will be given on different pre-trained CNNs and different edge detectors for filter selection.

3.2. Activation Peaks Extraction

Activation peaks represent the locations of key points within a feature map, which is in line with the input image. With the selected filters from every convolutional layer $\{f_n^* | n = 1, 2, \ldots, N\}$, the set of activation peaks $P_{f_n} = \left\{p^1, p^2, \ldots, p^k\right\}$ can be found by extracting local maxima from each feature map $f_n^*(I)$ using Algorithm 1 below.

As shown in Algorithm 1, assume a given threshold for repeat size is of ϕ pixels; the feature maps f_n must have a minimum size of 2ϕ, from which P_{f_n} is extracted. Let k be the required number of activation peaks. If the number of generated peaks is less than k, the size threshold is dynamically adjusted as $\phi = \phi - 1$ until the required number of generated peaks k is obtained. Next, the extracted activation peaks are mapped from feature space to image space to locate the pixel coordinates by rescaling, and an example is shown in Figure 2b.

Algorithm 1: Activation Peaks Selection

Input: Feature map $f_n^*(I)$, required number of activation peaks k, distance threshold ϕ
Output: Extracted set of activation peaks P_{f_n}

1. $P_{f_n} \leftarrow$ sort_descending(local_max($f_n^*(I)$), k)
2. **while** length(P_{f_n}) $< k$ **do**
3. $\quad \phi \leftarrow \phi - 1$
4. $\quad P_{f_n} \leftarrow$ sort_descending(local_max($f_n^*(I)$), ϕ)
5. **end while**
6. **return** P_{f_n}

3.3. Consistent Displacement Vector Selection

From the above step, a set of activation peaks are extracted. The most probable size of the repeated pattern of the image can be inferred as the most frequent distance between activation peaks in both image and feature spaces. To detect the repeated patterns from the activation peaks, all the vectors which link these peaks are calculated on the image space. Each vector then votes for the displacement it represents. The displacement which obtains the highest score then represents the diagonal of the detected repeating rectangle.

For each activated feature map $f_n^*(I)$, a set of displacement vectors $D_{f_n} = \left\{ d_{f_n}^{i,j} \right\}$ is obtained by computing the distance between each pair of peaks, i.e.,

$$d_{f_n}^{i,j} = \left| p^i - p^j \right|, \forall p^i, p^j \in P_{f_n}, i \neq j, \tag{3}$$

where $|.|$ denotes the element-wise absolute difference, namely the displacement of the two elements p^i, p^j. See Figure 3a for an illustration of some displacement vectors in the top sub-figure.

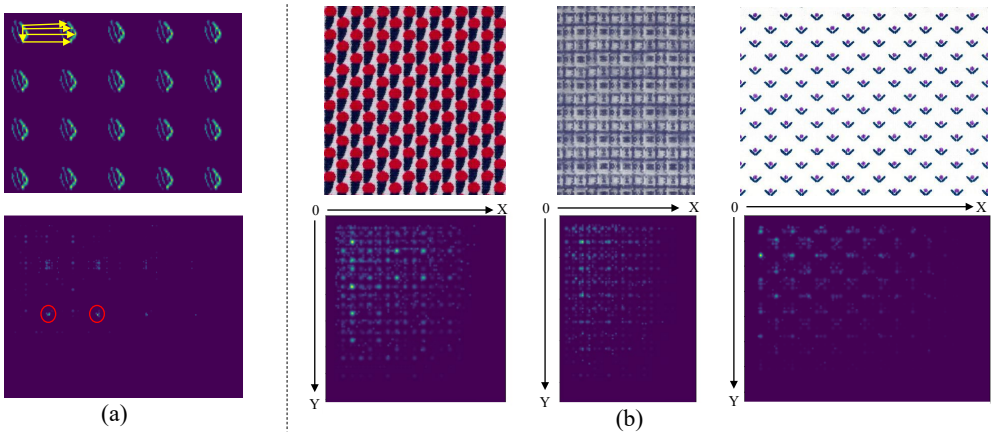

Figure 3. Illustration of consistent displacement vector selection. (**a**) Top: calculation of displacement vectors between activation peaks; bottom: voting space on which peaks' coordinates (light spots) correspond to the displacement vectors of the strongest repetitions; (**b**) more examples of the top row shows the input images and the bottom row shows the corresponding voting spaces.

Since each displacement vector $d_{f_n}^{i,j} \in D_{f_n}$ represents a candidate size of the repeated pattern, displacement vectors of all selected filters and all layers cast votes into Hough voting space V so as to estimate an initial repeated pattern size d^* (see Figure 3a bottom sub-figure for an example of voting space). To reflect the uncertainty of the localization of activation peaks, each displacement vector $V_{n,i,j}$ is modeled as normal distribution centers

at $d_{f_n}^{i,j}$ and standard deviation δ_n. To normalize overall energy, the vote is weighted by the number of displacement vectors $\left|D_{f_n}\right|$ across all layers and filters as follows:

$$V = \sum_{x,y} \sum_{n \in N} \frac{1}{\left|D_{f_n}\right|} \sum_{d_{f_n}^{i,j} \in D_{f_n}} V_{n,i,j} \tag{4a}$$

where

$$V_{n,i,j} = \frac{1}{2\pi\sqrt{|\Delta|}} exp(-\frac{1}{2}(x - d^{i,j})^T \Delta^{-1}(x - d^{i,j})) \tag{4b}$$

$$\Delta = \begin{pmatrix} \delta_n^2 & 0 \\ 0 & \delta_n^2 \end{pmatrix} \tag{4c}$$

where δ_n is related to the width w, corresponding to the x direction (or height h, corresponding to the y direction) of each feature map $w_{f_n^*(I)}$:

$$\delta_n = \frac{w}{2 \cdot w_{f_n^*(I)}} \tag{4d}$$

When the input image has high topology regularity, the repeated patterns can be represented as the most consistent displacement vector d^*, namely the maxima on the voting space along the x- and y-axes:

$$d^* = \left(d_x^*, d_y^*\right) = \left(\arg\max_x V_{x,0}, \arg\max_y V_{0,y}\right) \tag{5}$$

Figure 3b shows some examples of the voting space V, where the highlighted spots show the most consistent displacement vectors.

3.4. Repeated Pattern Size Optimization

Considering fronto-parallel images have the characteristics that even though with diverse image content, they have relatively regular or near-regular topology, an autocorrelation model integrating template matching with the *dynamic similarity threshold* is developed for repeated pattern size optimization in this study.

Based on the initial repeated pattern size d^*, a template patch T is obtained by cropping the image from the left-upper point $o = (0, 0)$ by size d^*. The autocorrelation model determines the repetition by matching template T on every possible area within the image by computing a pairwise correlation $cor_{(x,y)}$.

$$cor_{(x,y)} = \frac{\sum_{x,y}\left(T_{(0,0)} \cdot S_{(x,y)}\right)}{\sqrt{\sum_{x,y} {T_{(0,0)}}^2 \cdot \sum_{x,y} {S_{(x,y)}}^2}} \tag{6}$$

where $S_{(x,y)}$ denote the area started from point $(x,\ y)$, and the correlation $cor_{(x,y)}$ indicates the similarity degree between template T and area $S_{(x,y)}$. The larger correlation value $cor_{(x,y)}$, the higher the similarity.

Since the image can be understood as a continuously duplicating repeated pattern,the repetition structures in the image can be obtained by finding all the areas identical to the template patch. For efficient calculation and to avoid confusion at the starting region, the search is applied to a cropped image $I_{ox} \leftarrow I\left[\frac{1}{3}d_x^* : w, 0 : d_y^*\right]$, $I_{oy} \leftarrow I\left[0 : d_x^*, \frac{1}{3}d_y^* : h\right]$ instead of the entire image I. In order to optimize d_x^*, T is moved over the cropped image I_{ox}, pixel by pixel, and all the cor^x values are calculated by Equation (6).

Considering that input images may have diverse content, a dynamic threshold is set for identifying repeats:

$$sim_x = \varepsilon * \ max(cor^x) \tag{7}$$

where ε is adjustment parameter in range of $(0, 1)$. When the first matched area is found at the x^*th pixel in the x direction, i.e., $cor^{x^*} > sim_x$, the repeated pattern size in the x direction is refined as

$$d_x^{**} = x^* + \frac{1}{3} d_x^* \tag{8a}$$

Moreover, a minimum pixel distance, e.g., 2 pixels, is defined to avoid too small repeat sizes being detected. If the distance $\triangle d$ between the adjacent area $S_{w,0}$ and $S_{w+\triangle d,0}$ is smaller than the required minimum distance, then $d_x^{**} = d_x^{**} + \triangle d$. The threshold of pixel distance $\triangle d$ can be adjusted based on the complexity of the image. Similarly, the optimization is repeated along the y direction.

$$d_y^{**} = y^* + \frac{1}{3} d_y^* \tag{8b}$$

Figure 4a illustrates the size optimization along the x direction, and the repeated pattern grids before and after size optimization are compared in Figure 4b,c. As shown, the detection accuracy is significantly improved with this optimization algorithm.

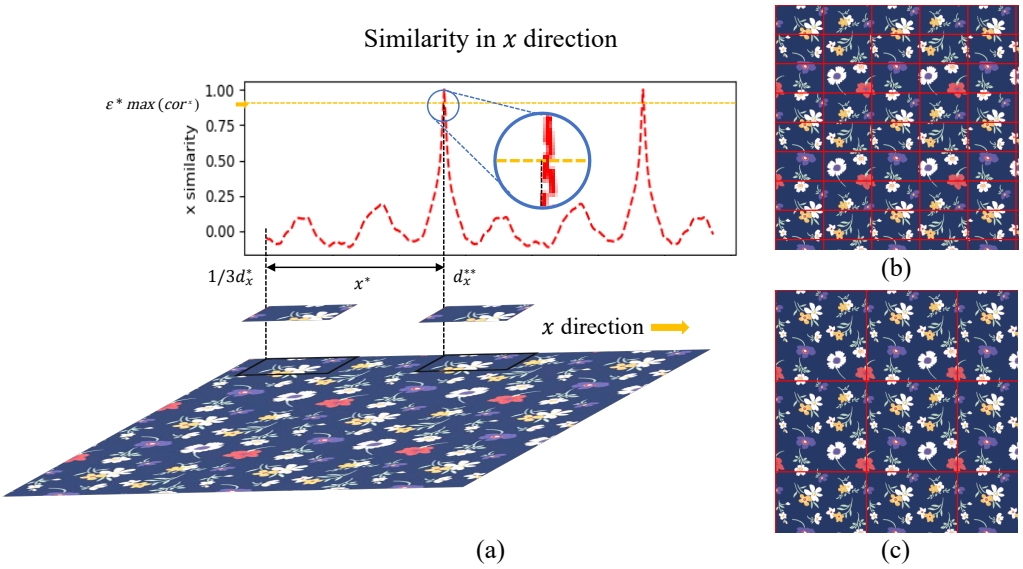

Figure 4. (**a**) An illustration of how the optimization algorithm works in the x direction. The correlation value changes when the template patch moves along the x direction and meets the minimum pixel distance. (**b**) Repeated pattern grid before optimization; (**c**) repeated pattern grid after optimization.

3.5. Repeated Pattern Grid Extraction

The output from the proposed method is a grid showing the location and size of the repeated patterns presented on the input images. The refined pattern sizes d^{**} obtained from the previous optimization step are used to decide the final grid.

To avoid $d^{i,j}$ being very different from d^{**}, the given $d^{i,j}$ is further screened by

$$D_{f_n}^* = \left\{ d_{f_n}^{i,j} \in D_{f_n} : ||d_{f_n}^{i,j} - d^{**}|| < 3\sigma \right\} \tag{9}$$

where parameter σ describes the radius of the neighborhood, showing the consistency of the displacement vectors for each filter/layer f_n.

Next, a centroid $c^* = (c_x, c_y)$ is extracted for the pattern by gathering and projecting the selected displacement vector votes $d_{f_n}^{i,j} \in D_{f_n}^*$ into a modular space (Figure 5a), and the centroid of the rectangle of these activations is calculated as follows:

$$c^* = \arg\min_{c} \sum_{\substack{n \in N \\ f_n^* \in F_n \\ d_{f_n}^{i,j} \in D_{f_n}^*}} W_{i,\,j,f_n^*} \cdot \left|\left| \mathcal{M}\left(d_{f_n}^{i,j} - c\right) - d^{**}/2 \right|\right| \tag{10}$$

where $\mathcal{M}(\cdot)$ converts the vector into modular space $\mathcal{M} : R^2 \rightarrow R$. In Equation (10), the votes are weighted by $W_{i,\,j,f_n^*}$, the strength of the activations, which is calculated as follows:

$$W_{i,\,j,f_n^*} = \frac{1}{\left|D_{f_n}^*\right| + \lambda} \cdot exp\left(-\frac{\left|\left| d_{f_n}^{i,j} - d^{**} \right|\right|^2}{2\sigma^2} \right) \tag{11}$$

where the parameter λ is a prior on the number of expected repetitions and is set to 0.8, following the same setting of [2,12]; to avoid biasing a certain layer over the others, σ is set to 1 to treat all layers equally.

Finally, the centroid is projected back into image space and an elastic repeated pattern grid is computed by a random sample consensus (RANSAC) algorithm [41] based on the identified centroids and optimized repeated size d^{**}; a few example results are shown in Figure 5b.

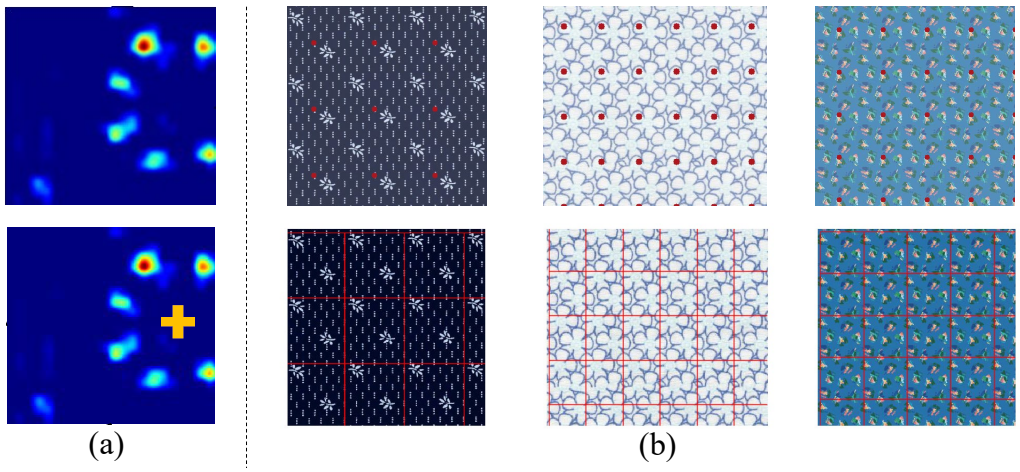

Figure 5. (**a**) Illustration for extraction of centroid: top—selected activation peaks projected to modular space, bottom—calculation of centroid. (**b**) Examples of centroids and elastic repeated pattern grids: top—identified centroids of repeated pattern, bottom—elastic repeated pattern grids calculated.

4. Experiments

4.1. Dataset and Evaluation Metric

There is no publicly available dataset of fronto-parallel images that can be used for quantitative analysis. To support the relevant study and evaluate the proposed method, a repeated pattern detection (RPD) dataset was constructed; the dataset has 841 images, including 774 scanned fabric images as well as 67 computer-generated design images. These images have a rich texture of diverse contents, with regular or near-regular topology, and the repeated pattern sizes of scanned fabric images were manually labeled as ground truth. Each image has more than two repeated patterns in its repeat directions. Although the RPD dataset is not a huge one, compared to other datasets for deep learning-based methods, the dataset is not used for training but for testing. Similarly to unsupervised learning or self-supervised learning, the optimization of repeat pattern size is performed

online based on every single image. A dataset of such size is sufficient for testing in terms of accuracy, robustness, and efficiency. The dataset (as well as the model) is shared publicly for comparative study (the url will be provided upon publication).

Since the output of the current method is represented in the form of a repeated pattern grid, instead of the location to a specific pixel position, the size of the repeated pattern grid is used to evaluate the accuracy of repeated pattern detection. Moreover, a repeated pattern may be found at any place in one image (as shown in Figure 6). Thus, repeated pattern grids were drawn from point $o = (0, 0)$ for the qualitative evaluation.

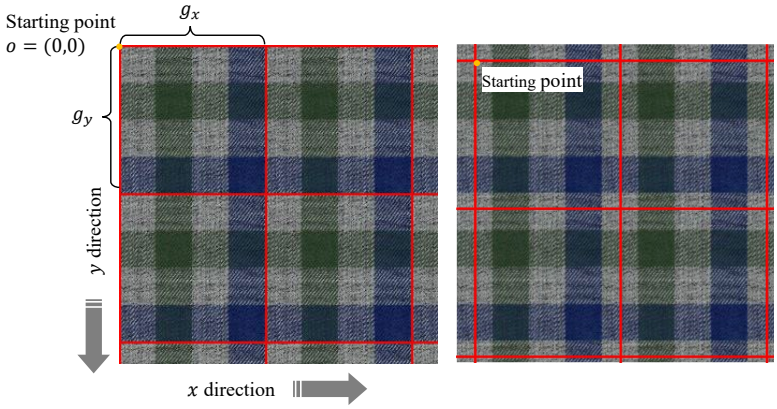

Figure 6. An example of ground truth image annotation. g_x and g_y are the ground truth in the x and y directions, respectively. The repeated pattern grid can be found from any point in the same image.

To evaluate the accuracy of each approach, it is assumed that all repeated patterns start from the same point and the intersection over union (IoU) is calculated to quantitatively measure the accuracy of the detected repeated pattern. The IoU is computed as follows:

$$IoU = \frac{\min(d_x^{**}, g_x) \cdot \min(d_y^{**}, g_y)}{d_x^{**} \cdot d_y^{**} + g_x \cdot g_y - \min(d_x^{**}, g_x) \cdot \min(d_y^{**}, g_y)} \tag{12}$$

where d_x^{**} and d_y^{**} are the sizes of the predicted repeated pattern in the x and y directions, respectively, and g_x and g_y are the ground truths for the minimal size of the repeated pattern.

4.2. Experimental Setting

The proposed repeated pattern detection method was developed on the PyTorch framework, and AlexNet [37] pre-trained on ImageNet was selected as the pre-trained CNN (orange box in Figure 2). The number of activation peaks of Algorithm 1 was set as $k = 20$, and the parameter ε from the dynamic threshold in Equation (7) in the repeated pattern size optimization method was set to 0.9.

4.3. Baseline and Selected Representative Work for Comparison Study

To objectively evaluate the proposed method, we compared it with five state-of-the-art (SOTA) methods, including two CNN-based methods [2,12] and three traditional methods [7,15,17]. The CNN-based methods by Lettry et al. [12] and Rodriguez-Pardo et al. [2] are the most relevant studies specifically designed for repeated pattern detection. The recent studies involving repeated pattern detection, such as those focusing on fabric defect detection [42] or landscape similarity analysis [29], are direct applications of Lettry et al.'s [12] method. Given the limited independent research on CNN-based repeated pattern detection algorithms, Lettry et al. [12] and Rodriguez-Pardo et al. [2] remain the most appropriate baselines for comparison. Furthermore, traditional methods [7,15,17] were included to ensure a comprehensive evaluation.

Lettry et al. [12] were the first to use activations in a pre-trained classification CNN to detect repeated patterns, and it is the baseline method of this study. They considered using almost all the CNN filters of the AlexNet.

Rodriguez-Pardo et al. [2] proposed to select filters of the AlexNet by abandoning filters whose activation values were less than 0.65 of the maximum activation value among all the filters in the same convolutional layer. Moreover, their method regularized input images in advance as a pre-processing step, which is time-consuming. An example of their pre-processing is shown in Figure 7. The left is their input image, with uneven yarns and non-uniform color variations, and the right is the regularized image [2]. In the proposed method, pre-processing was unnecessary and therefore not included. Furthermore, their method returns multiple repeated pattern sizes, instead of an optimized pattern size for each image; the size that is closest to the ground truth is chosen as the final result for comparison study.

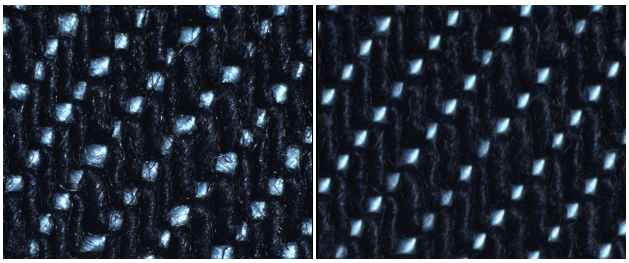

Figure 7. Example of pre-processing for image regularization [2].

A SIFT descriptor [7] is a classic local feature descriptor and is widely used in repeated pattern detection. Here, the same voting method (described in Equation (5)) was used to predict the repeated pattern sizes using a SIFT descriptor.

Neupane et al. [15] did not require a definition of template patches in advance. They copied the input image and cyclic-rotated it by one pixel each time; the cyclic-rotated direction of the repeated pattern was obtained by comparing and minimizing bitwise XOR differences between the input image and the cyclic-rotated one. The repeated pattern size was determined by performing these operations along the x and y directions of the image.

Comparatively, Qayum and Naseer [17] used template matching to predict the size of the repeated pattern of the input image. Qayum and Naseer [17] used a fixed initial point to search for repeat pattern size d^* for exact matching; in other words, there is no *dynamic similarity threshold* (ε) in Equation (7) or any flexible adjustment of pixel distance threshold $\triangle d$.

4.4. Experiments' Results

The proposed method was tested on the RPD dataset. Accuracy, efficiency, and robustness were evaluated in comparison to both SOTA traditional and deep learning-based methods. Both qualitative and quantitative analyses were conducted.

4.4.1. Qualitative Evaluation

For qualitative evaluation, the repeated pattern grids generated by the proposed method and other SOTA methods are presented in Figures 8 and 9, respectively, for scanned images of actual fabrics and computer-generated design images. Fabric images are typical fronto-parallel images with repeated patterns; most of them have rich texture information (see Figure 8). With the proposed method, repeated patterns were perfectly detected on these images, especially fabric images with dense patterns (columns 2 and 3 in Figure 8). The proposed method also performed well on fabric images with multiple design elements (columns 5 and 6 in Figure 8). Moreover, the image in column 7 was with many small

wrinkles and shadows, and an accurate detection result was obtained by the proposed method without pre-processing. It demonstrated the robustness of the proposed method.

Figure 8. Qualitative comparison for processing *actual fabric scan* images (along different columns) from results of (**a**) Qayum and Naseer [17], (**b**) Neupane et al. [15], (**c**) Rodriguez-Pardo et al. [2], (**d**) Lettry et al. [12], (**e**) SIFT [7], (**f**) the ground truth, and (**g**) the proposed method, respectively.

Lettry et al. [12] (row (d) in Figure 8) leveraged almost all the CNN filters' information, and acceptable results were achieved on most of the images. Although their detection results are not exactly the minimal repeated patterns (see columns 2, 3, 4, and 7 in Figure 8), they are indeed repetitive patterns that can be tiled to recover the input images. Furthermore, their detection results are more accurate than those of [2] (row (c) in Figure 8) and the approach leveraging SIFT [7] (row (e) in Figure 8). The main difference between Rodriguez-Pardo's and Lettry's work is that Rodriguez-Pardo introduced filter selection while Lettry had used all the filter results. As a result, about one-third less filters were used in Rodriguez-Pardo et al.'s [2] model than that in Lettry et al.'s [12], and comparatively less accurate detection results were generated. SIFT is a local image feature descriptor that does not consider global or hierarchical image information, and the results are less satisfactory in most of the example images. On the other hand, Qayum and Naseer [17] and Neupane et al. [15] had the worse results. In particular, Qayum and Naseer [17] only leveraged template matching, they set a relatively fixed size for template patches and chose the highest similarity value as the threshold on the x and y directions, respectively, and thus their method was less robust and easily confused by image texture. The results also demonstrate the superiority of deep feature-based methods in repeated pattern detection.

Computer-generated design images are almost noise-free images. Although some have complex design content, most methods performed better in this category than actual fabric

scan images (see Figure 9). Comparatively, the performance of Lettry's, Rodriguez-Pardo's, and SIFT-based methods degraded for images with a large number of small and similar design elements, namely motif patterns (see columns 2 to 4 in Figure 9), but were reasonably good for check designs (columns 5–7). The detection results of [15,17] in Figure 9 again were unsatisfactory as they could only detect relatively simple patterns and the output repeat sizes were not the smallest. For the computer-generated design images, the proposed method again outperformed all other methods.

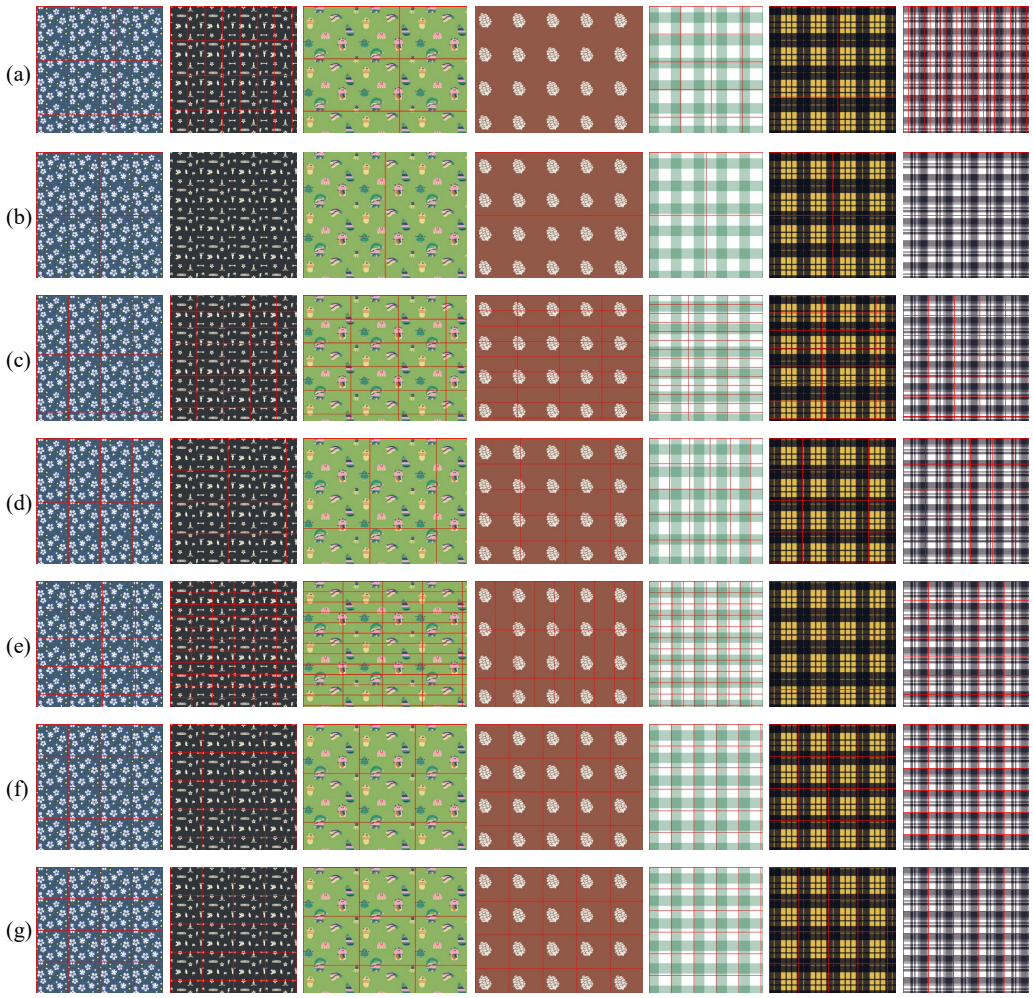

Figure 9. Qualitative comparison on *computer-generated* images (along different columns) from results of (**a**) Qayum and Naseer [17], (**b**) Neupane et al. [15], (**c**) Rodriguez-Pardo et al. [2], (**d**) Lettry et al. [12], (**e**) SIFT [7], (**f**) the ground truth, and (**g**) the proposed method, respectively.

4.4.2. Quantitative Evaluation

A quantitative evaluation using the RPD dataset, recording the computed time cost and resulting detection accuracies of various methods, is summarized in Table 1. Again, in the table, quantitative analysis was separated into two subsets of data: computer-generated images and actual fabric scan images. The table shows that the current work achieved higher accuracy than the other SOTA methods. The overall accuracy of the proposed method was the highest (67.3% on average) among all the methods, and so was the subsets' performance: the average accuracy of actual fabric scan images was 65.8% and 84.4% for the computer-generated images. Compared to the baseline method [12], the accuracy of the current method was 20% higher and the time cost was only 11% of the baseline.

Table 1. Summary of quantitative results: accuracy and time cost over different subsets of RPD dataset.

Image Type	Neupane et al. [15]		Qayum and Naseer [17]		SIFT-Based Approach [7]		Rodriguez-Pardo et al. [2]		Lettry et al. [12]		The Current Work	
	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc.(%)	Time (s)
Computer -generated	29.1	0.3	30.9	0.1	30.8	65.4	37.9	55.0	71.3	731.3	**84.4**	**46.3**
Scans of actual fabrics	11.6	0.2	16.0	0.1	33.5	61.0	40.9	49.0	54.3	335.8	**65.8**	**39.4**
All images	13.0	0.2	17.2	0.1	33.3	61.3	40.7	49.4	55.7	366.7	**67.3**	**40.0**

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 32 GB. **Bold** represents best performance.

Lettry et al.'s [12] average accuracy of actual fabric scanned images and computer-generated images were 54.3% and 71.3%, respectively, which are both higher than the corresponding results of [2]. Especially in the computer-generated category, the result of Lettry et al. [12] is nearly twice better than that of Rodriguez-Pardo et al. [2]. It can demonstrate the influence of the number of selected filters on the accuracy of the method; the less the filter is used, the lower the accuracy. It also justifies the reason for adding an optimization process in the current method. Moreover, traditional methods [7,15,17] perform poorly quantitatively on both real fabric and computer-generated images. Neupane et al. [15] performed the worst among all the methods.

Table 1 also shows the efficiency evaluation, namely the time cost, of all the methods, in addition to the accuracy performance (by Equation (12)). The current work is 40 s on average per image of the same input size, which is one-ninth of Lettry's time cost, about 10 s less than Rodriguez-Pardo's, and about 20 s less than the SIFT-based approach. Although Qayum and Naseer [17] and Neupane et al. [15] took a shorter time, their accuracies were very low. Therefore, the above experimental results have demonstrated that the current method design achieves a good balance between accuracy and efficiency.

To further analyze the efficiency of methods in Table 1, the per-step time costs of the proposed method (with respect to Figure 2) and other deep feature-based methods are compared in Table 2. Consistent displacement vector selection (step (c)) is the most time-consuming step of all the above deep feature-based methods. Displacement vector number and running time are positively correlated. In other words, the more filters that are used in a method, the more displacement vectors are created and thus the more time it will take in the voting process. As a result, the number of filters is negatively correlated to the final runtime. Consequently, by applying a filter selection algorithm, the proposed method achieved much better efficiency than that of [12].

Table 2. Per-step time cost (s) of all deep feature-based methods.

Methods	Filter Selection (a)	Activation Peaks Extraction (b)	Consistent Displacement Vector Selection (c)	Repeated Pattern Size Optimization (d) + (e)	Total
Rodriguez-Pardo et al. [2]	0.1	0.0	51.4	N/A	51.6
Lettry et al. [12]	0.1	36.1	327.1	N/A	363.2
The current work	5.9	0.0	33.9	0.0	40.0

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 32GB.

In addition to time cost, the computational expenses of the deep learning-based methods, Rodriguez-Pardo et al. [2] and Lettry et al. [12], and the current work, are further evaluated in terms of floating-point operations (FLOPs) and memory usage in Table 3. For these methods, all input images were resized to 224 × 224 pixels to ensure compatibility

with AlexNet. The FLOPs and memory usage were computed based on this standardized input size, providing a fair comparison of computational resources. The calculations follow the standard convolutional FLOPs formula [43], and memory usage estimations consider both model parameters and intermediate feature maps [44].

Table 3. Efficiency comparison between current and other deep-learning based methods.

Methods	Floating-Point Operations (FLOPs)	Memory Usage
Rodriguez-Pardo et al. [2]	0.48 G	40 MB
Lettry et al. [12]	0.72 G	60 MB
The current work	0.10 G	10 MB

Computer config: CPU Intel e5-2673 v3, GPU GeForce RTX 3060. Evaluate using input image sized 224 × 224.

For Rodriguez-Pardo et al. [2], who utilized approximately two-thirds of AlexNet's feature maps, the total FLOPs and memory usage were estimated at 0.48 GFLOPs and 40 MB, respectively. Lettry et al. [12], leveraging all feature maps of AlexNet, required 0.72 GFLOPs and 60 MB. In contrast, the current method, which processes only one feature map per convolutional layer in AlexNet, significantly reduced the computational burden to 0.1 GFLOPs and 10 MB while maintaining superior accuracy and efficiency. These results highlight the effectiveness of the proposed feature map selection strategy in reducing computational and memory costs.

It is worth noting that traditional methods, Qayum and Naseer [17], Neupane et al. [15], and the SIFT-based approach [7], process images at their original resolutions without resizing. As their computational burden depends on the image size, no direct comparison of FLOPs and memory usage is provided. However, the lack of feature map selection or optimization strategies in traditional methods suggests that their computational efficiency is generally lower than that of the deep learning-based approaches.

5. Ablation Study and Discussions

5.1. Ablation Study

To further examine the effectiveness of the feature processing and the optimization algorithm of the proposed method, an ablation study was conducted and reported in Table 4. As shown, the accuracy of any deep feature-based method is higher than that of the SIFT-based approach, including the proposed method without optimization. It proves that the selected filters still have the ability to extract useful features comparable to local feature descriptors. Moreover, the accuracy of the proposed method without an optimization module is similar to that of Rodriguez-Pardo but with slightly less time cost. It means that the designed filter selection method using edge results can substantially lower the computational cost yet maintain reasonable features for repeated pattern detection. Furthermore, the time cost with and without optimization was almost the same (both were 40 s), demonstrating that the proposed optimization algorithm dramatically improves detection accuracy without increasing the time consumption.

Table 4. Quantitative comparison of the current work without applying the optimization algorithm and when another feature-based approach is being applied.

Image Type	SIFT-Based Approach [7]		The Current Work w/o Optim. (Steps (a)–(c))		Rodriguez-Pardo et al. [2]		The Current Work (Steps (a)–(e))	
	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc. (%)	Time (s)	Acc. (%)	Time (s)
Computer-generated	30.8	65.4	32.2	46.2	37.9	55.0	84.4	46.3
Scans of actual fabrics	33.5	61.0	33.7	39.2	40.9	49.0	65.8	39.4
All images	33.3	61.3	33.6	40.0	40.7	49.4	67.3	40.0

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 32 GB.

5.2. Impact of Filter Selection Scheme and Different CNN Models

AlexNet was used in the proposed method as a CNN backbone for feature extraction. To justify the filter selection scheme of the current method, more experiments were conducted on the RPD dataset, using all the filters of AlexNet (denoted as "AlexNet*"), namely without filter selection. Moreover, experiments were also conducted using two other CNN models commonly used for feature extraction, including VGG16 [44] and ResNet50 [43]. The VGG16 network has five convolutional blocks, each containing several convolutional layers. The filter from the last convolutional layer of each convolutional block was selected. Five filters were selected for subsequent use. The ResNet50 has five convolutional stages, and each contains one or several convolutional layers. Similarly, when the ResNet50 network was used, only one filter from the last layer of each convolutional stage was selected. Table 5 presents the number of filters selected in the above experiments and their corresponding feature map sizes.

Table 5. Detailed setup of the extended experiments.

	AlexNet*		**AlexNet**		**VGG16**		**ResNet50**	
Layers	**num_filter**	**size_fmap**	**num_filter**	**size_fmap**	**num_filter**	**size_fmap**	**num_filter**	**size_fmap**
conv1	96	55 × 55	1	55 × 55	1	224 × 224	1	112 × 112
conv2	256	27 × 27	1	27 × 27	1	112 × 112	1	56 × 56
conv3	384	13 × 13	1	13 × 13	1	56 × 56	1	28 × 28
conv4	384	13 × 13	1	13 × 13	1	28 × 28	1	14 × 14
conv5	256	13 × 13	1	13 × 13	1	14 × 14	1	7 × 7

Table 6 shows the quantitative results of adopting different CNN models. AlexNet* used more filters than AlexNet. Surprisely, AlexNet* had almost no improvement in the average detection accuracy, but its average detection time was nine times that of current method with an AlexNet backbone with edge-based filter selection. It is believed that the number of filters used may not be the key reason for accuracy improvement. It can be seen from Table 6, although it is commonly understood that deeper CNN models perform better in feature extraction, the accuracy of repeated pattern detection is not significantly improved using deeper CNN features. Considering the balance between accuracy and computation cost, a pre-trained AlexNet could provide rich enough features for repeated pattern detection. In addition, the time cost of VGG16 was longer than that of ResNet50. As the corresponding selected feature maps had larger sizes, more activation peaks were extracted, taking longer processing time. It is therefore believed that the key factor determining the efficiency of a method is the number of applied activation peaks rather than the number of filters.

Table 6. Performance comparisons of using a different number of filters and pre-trained CNNs.

Image Type	**AlexNet***		**AlexNet**		**VGG16**		**ResNet50**	
	Acc. (%)	**Time (s)**	**Acc. (%)**	**Time (s)**	**Acc. (%)**	**Time (s)**	**Acc. (%)**	**Time (s)**
Computer-generated	90.5	731.4	**84.4**	**46.3**	88.0	455.4	86.0	232.2
Scans of actual fabrics	62.1	363.3	**65.8**	**39.4**	65.3	96.3	65.5	72.1
All images	64.3	366.8	**67.3**	**40.0**	66.4	113.5	67.5	85.7

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 32 GB. **Bold** represents best performance.

5.3. Impact of Different Edge/Boundary Detection Methods

In the proposed method, a Canny edge detector [40] is chosen to provide binary edge results for filter selection; see the pink box in Figure 2. The use of Canny edge detection is motivated by its ability to extract object boundaries and visually salient edges that preserve the essential structure of the image while ignoring weakly related information [38,39].

Additionally, the binary nature of the Canny edge output makes it computationally efficient and straightforward to combine with feature maps. Specifically, the binary edge map ([0,1]) can be directly multiplied with feature maps, enabling the extraction of more relevant and effective information. Experiments were conducted to compare the performance of Canny with a learning-based edge detection method, BDCN [39], for filter selection and feature extraction. BDCN achieves superior performance in edge detection [39], as shown in Table 7, the overall accuracy of our method remained identical (67.3%) when using either Canny or BDCN. However, using BDCN significantly increased the computational time from 40.0 s (of Canny) to 44.7 s (of BDCN), making it less efficient for practical application. This is particularly evident for computer-generated images, where the runtime increased by over 40 s. These findings demonstrate that while BDCN achieved superior edge detection performance in isolation, it did not improve the accuracy of our method and added unnecessary computational overhead. Canny is, therefore, chosen in our method, considering both accuracy and efficiency.

Table 7. Summary of quantitative results of using different boundary detection methods.

Image Type	Alexnet + Canny [40]		Alexnet + BDCN [39]	
	Accuracy (%)	Overall Time (s)	Accuracy (%)	Overall Time (s)
Computer-generated	84.4	46.3	87.6	86.8
Scans of actual fabrics	65.8	39.4	65.6	41.0
All images	67.3	40.0	67.3	44.7

Computer config: CPU Intel i7-6700k, GPU GeForce GTX TITAN X, Memory 32 GB.

5.4. Practical Application as Image Pre-Processing for Textile Design

The proposed method serves as a general-purpose framework for repeated pattern detection, with the flexibility to detect a wide range of repeated patterns from images across various domains. In this paper, its practical value is demonstrated through an application to textile design. The workflow of textile design begins by processing a large input image containing multiple repeated patterns using the proposed method. This step detects and isolates the repeated patterns, trimming them into individual repeating units. These isolated patterns are then fed into an external design element extraction and vectorization method [45], which outputs vectorized design elements. This automated workflow streamlines the pre-processing stage, significantly reducing manual effort, simplifying input data, and ensuring consistency in isolating design elements for downstream tasks.

5.5. Experiments on Real-World Non-Textile Images

As discussed in the introduction, and also illustrated in Figure 1, repeated pattern images are everywhere, and they are different in terms of content diversity and topological regularity. The proposed method covering five steps/modules can be flexibly adjusted based on different target image types. It is interesting to explore whether the proposed method can also work on other fronto-parallel images with complex content and near-regular topology structure. Thus, the proposed method was tested on some real-world images, e.g., environmental images taken from the testing dataset of [12].

Figure 10 compares the results from Lettry's method and the current method, where the number of activation peaks for Algorithm 1 was set as $k = 20$; and the parameter ε in the optimization module (Equation (7)) was set to 0.87. It can be shown that although the current method uses fewer filters than that of Lettry, similar or better detection results are achieved on these environmental images. For example, the current method performed better on the fifth image that had a very strong illumination difference. It proves that the current method could achieve satisfactory results on fronto-parallel non-textile real-world images with significantly varied content by adjusting parameters. This supports

the conjecture that most fronto-parallel repeated patterns can be detected by tactically integrating the modules with different degrees of coupling.

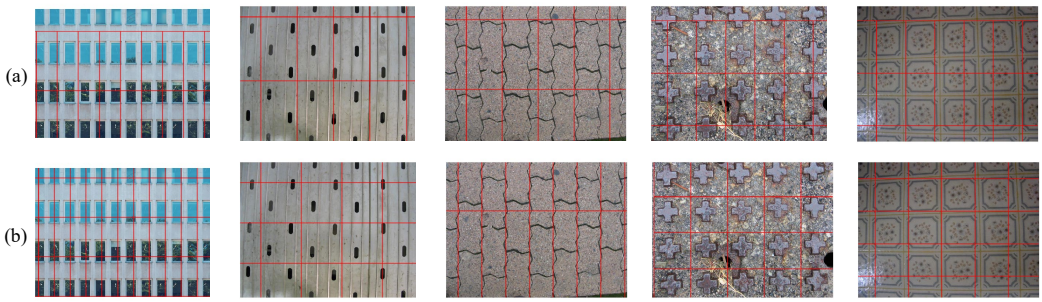

Figure 10. Detection results of environmental images: (**a**) the detection results from [12], and (**b**) the results from the current method.

5.6. Limitations

The proposed method has the following two limitations. First, the current feature fusion strategy, although it effectively balances efficiency and accuracy, limits the amount of deep feature information being utilized. Second, the method has been primarily evaluated on the RPD dataset, which focuses on textile images. While initial experiments on datasets such as building facades and floor tiles have shown encouraging results, further validation on more and different domains of images is necessary to confirm its generalizability. Additionally, the current method assumes fronto-parallel images as input, and its performance may degrade under perspective distortions, occlusions, or irregular patterns. Extending the method to handle such challenges would broaden its applicability to more complex and diverse real-world scenarios.

6. Conclusions

In this paper, a hybrid method has been proposed for repeated pattern detection on unknown fronto-parallel images. Unlike previous methods that rely solely on image features or prediction models, the proposed method combines deep features and an autocorrelation model to achieve a good balance between accuracy and efficiency. Specifically, the method introduces a filter selection strategy based on edge detection results to extract the most effective filters for feature extraction. Furthermore, a dynamic similarity threshold has been incorporated into the autocorrelation model for template matching, enabling more accurate repeated pattern size optimization. A dataset (RPD) containing 774 real fabric scanned images and 61 computer-generated images has been built and shared for comparative study. Experimental results on the RPD dataset have demonstrated that the proposed method achieved the highest overall accuracy (67.3%), with 65.8% accuracy on real fabric images and 84.4% on computer-generated images. The average runtime was 40 s per image, which was only 11% of the runtime of the baseline method, while requiring just 0.1 GFLOPs and 10 MB of memory, compared to 0.72 GFLOPs and 60 MB for the baseline method. These results illustrate the effectiveness of the proposed filter selection strategy, resulting in the best balance between accuracy and efficiency.

Further experiments have been conducted to analyze the contributions of different components in the proposed method. Replacing AlexNet with deeper CNN models such as VGG16 and ResNet50 showed that deeper networks do not significantly improve accuracy when the number of selected filters is fixed, but they substantially increase runtime. Similarly, replacing the Canny edge detector with a deep learning-based boundary detector (BDCN) does not lead to significant accuracy improvement while slightly increasing the time costs. These results validate that the proposed method achieves an effective trade-off

between accuracy and efficiency by carefully balancing feature selection and computational complexity. In addition, the proposed method has demonstrated good adaptability and practical value in real-world applications, such as design element extraction and resource preparation in textile design. Beyond textiles, the method has been shown to be able to detect diverse repeated patterns, highlighting its potential for broader applications, such as defect detection and texture-based image retrieval.

In the future, further work should focus on developing a better fusion algorithm for consistent displacement vector selection from selected feature maps, allowing more activation peaks to be considered for improved accuracy. Moreover, a more advanced backbone network could be explored to further enhance both accuracy and efficiency. Beyond this, another key direction is to extend the application of repeated pattern detection results to other tasks, such as fabric defect detection, texture-based image retrieval, and structural pattern analysis in real-world scenarios. These efforts will broaden the impact of repeated pattern detection and explore its potential in various application fields.

Author Contributions: Conceptualization, Y.Z. and P.Y.M.; Methodology, H.Q. and Y.Z.; Software, H.Q. and G.F.; Formal analysis, H.Q. and Y.Z.; Investigation, H.Q.; Resources, G.F.; Writing—original draft, H.Q.; Writing—review and editing, P.Y.M. and L.L.; Visualization, L.L.; Supervision, P.Y.M. and L.L.; Funding acquisition, P.Y.M. and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: The work described in this paper was supported by grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project Nos. 152112/19E and 15602323). The work was also supported, in part, by The Hong Kong Polytechnic University (Project Nos. P0049355/CD95 and P0051330/BDVH).

Data Availability Statement: The original data presented in the study are openly available at Github https://github.com/ZoeQU/Efficient-and-Effective-Detection-of-Repeated-Patterns.git (accessed on 22 January 2025).

Conflicts of Interest: Author Gerhard Flatz is employed by the KTC Limited. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Lin, W.C.; Hays, J.; Wu, C.; Liu, Y.; Kwatra, V. Quantitative evaluation of near regular texture synthesis algorithms. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 17–22 June 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 427–434. [CrossRef]
2. Rodriguez-Pardo, C.; Suja, S.; Pascual, D.; Lopez-Moreno, J.; Garces, E. Automatic extraction and synthesis of regular repeatable patterns. *Comput. Graph.* **2019**, *83*, 33–41. [CrossRef]
3. Tao, C.; Zhou, J.; Yin, M. Automatic identification of textile pattern consecutiveness based on similarity space. *Text. Res. J.* **2017**, *87*, 224–231. [CrossRef]
4. Zhang, J.; Pan, R.; Gao, W.; Zhu, D. Automatic recognition of the color effect of yarn-dyed fabric by the smallest repeat unit recognition algorithm. *Text. Res. J.* **2015**, *85*, 432–446. [CrossRef]
5. Liu, Y.; Collins, R.T.; Tsin, Y. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 354–371. [CrossRef] [PubMed]
6. Schindler, G.; Krishnamurthy, P.; Lublinerman, R.; Liu, Y.; Dellaert, F. Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–7. [CrossRef]
7. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; Volume 2, pp. 1150–1157. [CrossRef]
8. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; IEEE: Piscataway, NJ, USA, 2005; Volume 1, pp. 886–893. [CrossRef]

9. Pfeffer, M.A.; Ling, S.H. Evolving Optimised Convolutional Neural Networks for Lung Cancer Classification. *Signals* **2022**, *3*, 284–295. [CrossRef]
10. Laganà, F.; Pratticò, D.; Angiulli, G.; Oliva, G.; Pullano, S.A.; Versaci, M.; Foresta, F.L. Development of an Integrated System of sEMG Signal Acquisition, Processing, and Analysis with AI Techniques. *Signals* **2024**, *5*, 476–493. [CrossRef]
11. Rezapour, M.; Ksaibati, K. Convolutional Neural Network for Roadside Barriers Detection: Transfer Learning versus Non-Transfer Learning. *Signals* **2021**, *2*, 72–86. [CrossRef]
12. Lettry, L.; Perdoch, M.; Vanhoey, K.; Gool, L.V. Repeated pattern detection using CNN activations. In Proceedings of the 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, USA, 24–31 March 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 47–55. [CrossRef]
13. Doubek, P.; Matas, J.; Perdoch, M.; Chum, O. Image matching and retrieval by repetitive patterns. In Proceedings of the 2010 20th International Conference on Pattern Recognition, Istanbul, Turkey, 23–26 August 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 3195–3198. [CrossRef]
14. Lin, H.C.; Wang, L.L.; Yang, S.N. Extracting periodicity of a regular texture based on autocorrelation functions. *Pattern Recognit. Lett.* **1997**, *18*, 433–443. [CrossRef]
15. Neupane, P.; Tuladhar, A.; Sharma, S.; Tamang, R. Extracting Unknown Repeated Pattern in Tiled Images. In Proceedings of the International Conference on Hybrid Intelligent Systems, Bhopal, India, 10–12 December 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 92–102. [CrossRef]
16. Harel, D. *Template Matching Techniques in Computer Vision: Theory and Practice*; Wiley: Hoboken, NJ, USA, 2009.
17. Qayum, M.A.; Naseer, M. A fast approach for finding design repeat in textile rotary printing for fault detection. *J. Text. Inst.* **2017**, *108*, 62–65. [CrossRef]
18. Cai, Y.; Baciu, G. Detection of repetitive patterns in near regular texture images. In Proceedings of the 2011 IEEE 10th IVMSP Workshop: Perception and Visual Signal Analysis, Ithaca, NY, USA, 16–17 June 2011; IEEE: Berlin/Heidelberg, Germany, 2011; pp. 60–65. [CrossRef]
19. Kuo, C.; Shih, C.; Lee, J. Separating Color and Identifying Repeat Pattern Through the Automatic Computerized Analysis System for Printed Fabrics. *J. Inf. Sci. Eng.* **2008**, *24*, 453–467. [CrossRef]
20. Conners, R.W.; Harlow, C.A. Toward a structural textural analyzer based on statistical methods. *Comput. Graph. Image Process.* **1980**, *12*, 224–256. [CrossRef]
21. Pinho, A.J.; Ferreira, P.J. Finding unknown repeated patterns in images. In Proceedings of the 2011 19th European Signal Processing Conference, Barcelona, Spain, 29 August–2 September 2011; IEEE: Berlin/Heidelberg, Germany, 2011; pp. 584–588.
22. Tuytelaars, T.; Turina, A.; Gool, L.V. Noncombinatorial detection of regular repetitions under perspective skew. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 418–432. [CrossRef]
23. Liu, J.; Liu, Y. Grasp recurring patterns from a single view. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 2003–2010. [CrossRef]
24. Pritts, J.; Chum, O.; Matas, J. Detection, rectification and segmentation of coplanar repeated patterns. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 2973–2980. [CrossRef]
25. Guo, Y.; Ge, X.; Yu, M.; Yan, G.; Liu, Y. Automatic recognition method for the repeat size of a weave pattern on a woven fabric image. *Text. Res. J.* **2019**, *89*, 2754–2775. [CrossRef]
26. Liu, J.; Psarakis, E.; Feng, Y.; Stamos, I. A Kronecker Product Model for Repeated Pattern Detection on 2D Urban Images. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2266–2272. [CrossRef]
27. Ahmad, J.; Muhammad, K.; Baik, S.W. Medical image retrieval with compact binary codes generated in frequency domain using highly reactive convolutional features. *J. Med. Syst.* **2018**, *42*, 1–19. [CrossRef] [PubMed]
28. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning deep features for discriminative localization. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 2921–2929. [CrossRef]
29. Malik, K.; Robertson, C. Landscape Similarity Analysis Using Texture Encoded Deep-Learning Features on Unclassified Remote Sensing Imagery. *Remote Sens.* **2021**, *13*, 492. [CrossRef]
30. Long, J.L.; Zhang, N.; Darrell, T. Do convnets learn correspondence? In Proceedings of the 27th International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; MIT Press: Cambridge, MA, USA, 2014; pp. 1601–1609. [CrossRef]
31. Yang, B.; Yan, J.; Lei, Z.; Li, S.Z. Convolutional channel features. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 82–90. [CrossRef]
32. Amin, A.M.E.; Liu, Q.; Wang, Y. Convolutional neural network features based change detection in satellite images. In Proceedings of the First International Workshop on Pattern Recognition, Tokyo, Japan, 11–13 May 2016; SPIE: St Bellingham, WA, USA, 2016; pp. 181–186. [CrossRef]

33. Xiang, J.; Zhang, N.; Pan, R.; Gao, W. Fabric image retrieval system using hierarchical search based on deep convolutional neural network. *IEEE Access* **2019**, *7*, 35405–35417. [CrossRef]
34. Wang, Y.; Wei, X.; Luo, L.; Wen, W.; Wang, Y. Robust RGB-D tracking via compact CNN features. *Eng. Appl. Artif. Intell.* **2020**, *96*, 103974. [CrossRef]
35. Li, T.; Wu, B.; Yang, Y.; Fan, Y.; Zhang, Y.; Liu, W. Compressing convolutional neural networks via factorized convolutional filters. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3972–3981.
36. Xie, Y.; Xiao, J.; Huang, K.; Thiyagalingam, J.; Zhao, Y. Correlation filter selection for visual tracking using reinforcement learning. *IEEE Trans. Circuits Syst. Video Technol.* **2018**, *30*, 192–204. [CrossRef]
37. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
38. Ansari, M.A.; Kurchaniya, D.; Dixit, M. A comprehensive analysis of image edge detection techniques. *Int. J. Multimed. Ubiquitous Eng.* **2017**, *12*, 1–12. [CrossRef]
39. He, J.; Zhang, S.; Yang, M.; Shan, Y.; Huang, T. BDCN: Bi-directional cascade network for perceptual edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 100–113. [CrossRef] [PubMed]
40. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *6*, 679–698. [CrossRef]
41. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
42. Huang, Y.; Xiang, Z. RPDNet: Automatic Fabric Defect Detection Based on a Convolutional Neural Network and Repeated Pattern Analysis. *Sensors* **2022**, *22*, 6226. [CrossRef] [PubMed]
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
44. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556. [CrossRef]
45. Qu, H.; Zhou, Y.; Chau, K.; Mok, P. Design elements extraction based on unsupervised segmentation and compact vectorization. In Proceedings of the 16th International Conference on Computer Graphics, Visualization, Computer Vision and Image Processing, CGVCVIP 2022, Lisbon, Portugal, 20–22 July 2022; IADIS Press: Lisbon, Portugal, 2022; pp. 78–84.